Network Management as a Service

Christos Gkantsidis and Hitesh Ballani
Microsoft Research
Cambridge, UK

Microsoft Research MSR-TR-2010-83

Abstract

This paper explores the feasibility of offering network management as a service. We describe availability and responsiveness as the two key factors that govern how functionality is moved off-site (to the cloud). Using common management tasks as examples, we describe the process and the challenges of designing their cloud-based equivalents. We also examine the costs of such off-site implementations.

1 Introduction

Network management is hard. This is especially true for small to medium size networks where even basic management operations require inordinate manual effort. Automated management tools, when available, require significant human expertise to operate, not to mention the cost of implementing and supporting them. Consequently, the deployment and use of sophisticated tools is rare in all but a few very large ISPs and enterprises [2]. The functionality offered by small and medium size networks is thus held ransom by the owner's ability to cope with the management complexity.

Similar concerns have plagued small-to-medium sized entities in regards to their ability to maintain infrastructure such as email, storage and even compute infrastructure in a cost-effective fashion. We note that these concerns have prompted a move towards the "cloud" with more and more infrastructure as a service offerings. In this context, we ask the following question:

> *Is it possible to offer network management as a service?*

Such off-site network management, if feasible, would benefit from the amortization of hardware, software and human costs with network owners having the flexibility of paying for what they use. Shared infrastructure also opens the door for management functionality which otherwise might be considered too onerous to implement and maintain. For instance, few networks today have the ability to test configuration changes before applying them to a live network. It might be more practical to offer an "on-demand network lab" service that can emulate client networks and allow them to test configuration changes. Further, offering network management as a service would allow management software to be maintained in a centralized fashion by experienced individuals and would encourage good practices. Finally, such consolidation is more conducive to the management plane keeping up with data plane evolution which is very difficult with today's setup [3].

While the primary benefit of off-site network management is cost savings, it has technical advantages too. Cloud providers managing several networks would have information feeds like network topology, traffic patterns, etc. from the managed networks. This allows for the use of learning algorithms for management problems; for instance, debugging faults ([1,9] exemplify such an approach), attack prevention, and network planning tasks seem particularly suited to utilize the extra information.

However, there are many challenges in the way of practical management of networks from the cloud. Two factors differentiate such management from the status quo: (i). the management plane runs *remotely* which loads to obvious questions about its responsiveness, robustness, availability and security, and (ii). the management plane is operated by an entity other than the network owner which raises security and privacy concerns.

In this paper, we focus on the first challenge, i.e. the fact that the management plane is not colocated with data plane devices, and show how it influences the design of cloud-based network management. To this effect, we classify network management tasks based on the degree to which they need to be *responsive* (accessible with low latency) and *available* (always accessible). We find that management tasks that don't have stringent responsiveness requirements fit naturally in the cloud management model. For tasks that are latency-sensitive, we propose embedding data plane devices with the minimum amount of functionality needed to mask the latency to the cloud management plane. We elaborate this through concrete management examples in

the rest of this paper. Further, we present a first cut analysis of the costs of offering management services from the cloud and show that it is not prohibitive.

Overall, we recognize that the thought of an external entity managing a network remotely sounds preposterous. However, one could argue that the same could have been said about the possibility of an enterprise relying on an external email service till only a few years ago. To this effect, we hope that the thought experiment presented in this paper shows that off-site network management is not as implausible as a first look might suggest.

2 Refactoring Functionality

Network management contains a very diverse set of functions, including among others planning, allocating, deploying, coordinating, and monitoring the resources of a network, traffic routing, configuration management, fault management, security management, and many others.[1] Today, the implementation of these functions is split between data-plane devices (running distributed instantiations of control protocols) and management applications in an ad-hoc fashion. This, in turn, impedes structured network management [8].

Consequently, a spate of recent proposals, pioneered by the 4D project [8,18], involve refactoring of management functionality by simplifying data plane devices so that centralized controllers can do a better job of managing the network. SANE [6],Ethane [5], OpenFlow [13], etc represent a few examples of such proposals. We begin with the thesis that the flexibility and power offered by such designs involving a logically centralized management plane is more amenable than the status quo to a remote management scenario where the applications managing the network reside in and are operated by cloud providers. Hence, the design in the rest of this paper starts off from the 4D model involving "thin" data-plane devices that inform the management plane of their basic connectivity and need to be fed with management state (example, routing tables) to operate.

However, we find that in some cases the remoteness of the management plane necessitates a different refactoring of functionality than what has been proposed in the past. In the following section, we use example network management tasks to illustrate this.

3 Network Management Functions

A remotely hosted management plane implies that the latency to access it will be higher (bounded at the bottom by the propagation latency to the cloud). Further, the external links connecting the network being managed to the cloud are part of the management plane but not the data-plane of the network per se. Thus, it is possible that the management plane is not accessible even when there are no failures in the managed network and in such cases, the data plane should operate correctly.

[1] http://en.wikipedia.org/wiki/Network_management

Availability \ Responsiveness	flexible	stringent
flexible	Natural fit eg, Traffic Engineering (section 3.1)	-
stringent	Natural Fit eg, Active Directory (section 3.2)	Challenging eg, VPN, routing (section 3.3)

Table 1: Management functions offered remotely

To show the feasibility of cloud-based management in the face of these challenges, we divide management functions according to two criteria. The first is the expected responsiveness of the function which, in turn, is the latency to access the function. Routing for examples requires fast response times in order to re-route around failed links and nodes. On the other hand, capacity planning and other traffic engineering activities do not typically have such stringent requirements.

The second criterion is availability, i.e. the dependence the users have on being able to access and use the service. Responsiveness requires availability, the reverse however is not necessary. For example, an active directory service that is used to achieve access control in a network should always be available. However, the increase in latency if active directory servers were to be hosted remotely may not hurt the users since they need to contact the servers only at connection time and the resulting authorization is cached for future use.

To separate these two criteria, the rest of this section naively assume that a cloud management service offers good availability (We discuss the validity of this assumption in section 4.2.) Given this, Table 1 summarizes how different management functions can be offered remotely. Activities that do not have stringent responsiveness requirements fit naturally and can be moved to the cloud with ease. Activities that do need to be responsive are more challenging and require refactoring of functionality so as to mask the latency to the cloud.

3.1 Functions with flexible responsiveness and availability requirements

Management tasks that do not depend critically on responsiveness and availability are obvious candidates to be implemented as a service. We classify such tasks into two categories based on what a cloud implementation has to offer.

– *Cost benefits for infrequently run tasks.* A fair fraction of management tasks are executed infrequently. For instance, traffic engineering tasks such as optimization of link weights based on traffic matrices are aimed to account for changes in traffic patterns over periods ranging from hours to days to weeks and longer. Checking network configuration to detect misconfigurations [4], or even testing configuration changes before applying them are two other examples.[2] For such infrequent activities, the cost of acquiring the necessary software and experience may be prohibitive for many networks.

On the other hand, existing designs for such operations can be trivially extended to run remotely. We show two examples in Fig. 1. As shown in the figure, today's link weight optimization services can be hosted in the cloud and shared by multiple networks leading to amortized costs. An easily accessible offering of link weight management and other similar services may even encourage network managers with limited experience or resources to use optimization as part of their network management activities.

– *Tasks enhanced by shared information.* Beyond cost savings, even the efficacy of many management tasks can be enhanced by utilizing information from other networks and the cloud provides a good avenue to do so. As a matter of fact, there has been a recent spurt in interest in the use of shared information as a means of detecting, debugging and diagnosing problems. NetPrints [1] and NetMedic [9] represent prominent examples of the same. The basic idea here is to maintain a common repository of configurations and problems, and to mine that information in a automated fashion in order to detect the root cause of a problem. Specifically, these systems can be used to identify the common patterns observed by many networks that experience the same problems and use this to diagnose the problem.

In essence, both systems use the power of the crowd to perform more advanced debugging. Given their reliance on information from client networks, we argue that these are working examples of cloud-based management. A similar approach can be applied to tackling planning problems and answering "what-if" scenarios [17]. For instance, a cloud service can use its knowledge about how its client networks have evolved to predict the impact of capacity addition for a specific client network. Today, such capacity planning is mostly an ad-hoc process relying on knowledge embedded inside human experts, if any, managing the client network.

Note that unlike the aforementioned class of infrequent activities which may be cheaper to implement as a cloud service, but where the central server can be part of the network (and not necessarily reside remotely), information sharing tasks do require a centralized cloud service, where many operators can upload configuration information, errors and other information. Also, unlike the previous examples, where the service could have been implemented strictly as a processing service and leave the ownership of the data to the network operator (i.e., the processing elements can throw away the data after they finish processing), in this class of activities, network configuration (and mis-configuration) information is stored externally for a potentially long time period. This approach raises such privacy and security concerns; however, we note that a range of businesses value such risks lower than the expected benefits.

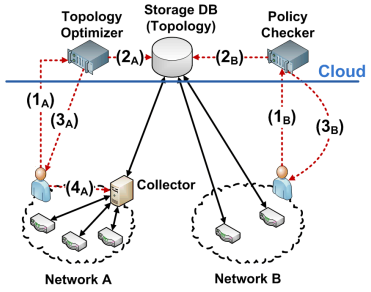

Figure 1: Examples of network planning and other optimization services. Central to this example is a storage server that collects information about the network. The information is collected either with the help of a "collector" (left), or directly from the cloud-enabled network elements (right). The manager on the left wishes to optimize the routing process in her network. (1_A) She contacts the relevant cloud service, which (2_A) retrieves from the storage server the topology and the traffic demands, computes the optimal solution, and (3_A) communicates the results to the operator who (4_A) initiates, through the collector, a series of configuration updates. The manager on the right wishes to check whether the configuration satisfies the policy constraints of the network. He contacts a policy checker service, which reads the related information from the storage server, verifies the configuration, and returns the results to the manager.

[2]For example, virtualization technologies, such as those described in `http://www.ipflow.utc.fr/index.php/Cisco_7200_Simulator` and `http://www.gns3.net/` can be used to simulate in a controlled environment the control plane of the network using potentially the same software and configuration as those used in the live network. Such a network lab is easier to implement and maintain as a shared resource.

3.2 Functions that depend on availability, but are flexible on responsiveness

For some management functions, their design relaxes responsiveness requirements. A good example is services relying on caching. Management functions can use client-side caching to relax the need for responsiveness. Consequently, the bar for moving such functionality to the cloud is low because a remote implementation will not degrade responsiveness. Indeed, there have been many examples that have shown that such transition is possible (and beneficial to the users), including Email hosted services, Customer Relationship Management services, and Intranet Web services. Of course, the functions will still need to offer high availability and section 4.2 discusses ways to achieve this.

A concrete example of such a management function is directory services that are often used to manage authentication, authorization and even information dissemination in enterprise networks. Specifically, Active Directory is a directory service used for controlling user access to machines, services, and resources, and for storing site-specific information. Typically, human managers implement this service by provisioning and maintaining logically centralized Active Directory servers that serve clients. However, clients cache responses (query responses, leases, credentials, etc.) so that they don't need to access the servers for every network operation. This implies that moving the directory servers to the cloud, as shown in Figure 2, will probably have little to no impact noticeable to the client. Further, as shown later, the monetary costs of such a move are small.

3.3 Functions that require responsiveness

Management functions that have stringent responsiveness requirements cannot be moved to the cloud as is. Instead they need to be refactored into two components: functionality that is latency sensitive and hence, should reside on data plane devices and functionality that can reside in the cloud.

Lets consider the example of Virtual Private Networks (VPNs) used to provide network access to mobile users. VPN set up involves a mobile user contacting a VPN server, which first validates the identity of the user, checks for compliance with network's policies (e.g. check for the latest version of the antivirus and other required software patches), and then establishes a bidirectional tunnel. As shown in Fig. 3, a cloud-based VPN service would require that the first two steps, i.e. authentication and policy checking, be implemented in the cloud whereas the actual tunneling/detunneling be implemented by the network. This would reduce the complexity of managing VPN access by "centralizing" checking and authentication, especially for geo-distributed client networks. Even the effort of knowing about the latest software patches and virus definitions will be offloaded. Moreover, the cloud has better view of the load on the entrance points and their location, and may make a better decision on allocating users to best entry points (in many current solutions, it is the user that typically makes that choice).

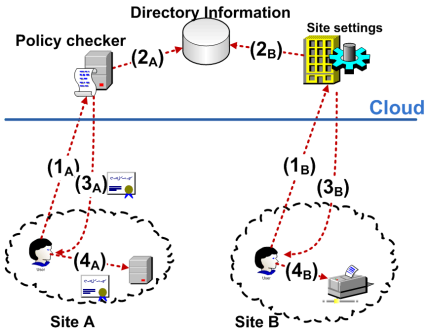

Figure 2: Two examples of how directory services can be moved to the cloud. On the left, a user wishing to access a internal web server, first contacts the cloud policy checker, which consults the database that contains site authentication and authorization information. Then, the checker issues a certificate that can be used by the user to access the server. On the right, a user wishes to find nearby printers. She consults a server that stores site specific information, who responds with information about the printer.

We reiterate that unlike the previous case, where we proposed to move the entire directory service to the cloud, in this example we propose to decompose the VPN service into components and offload to the cloud only those components that are in the control path, but not on the data path. We believe that this approach may be necessary for other services that want to take advantage of the cloud, but cannot afford the latency of sending all traffic to the cloud (in this case, we wanted to avoid sending all packets first to the cloud and then to the network).

4 Challenges using cloud network management

We now discuss two main challenges that may discourage the adoption of network management as a service: cost and availability. We show that many of the services described above have small operational costs. We examine the factors that affect availability, and outline approaches that can improve it.

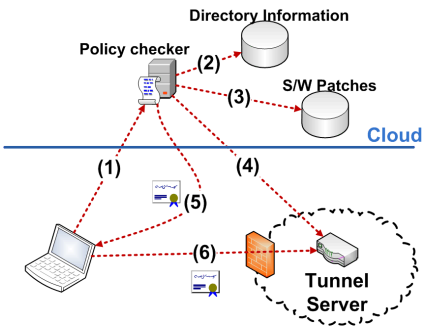

Figure 3: An example of providing authentication, authorization, and policy control by the cloud, in order to allow a user to VPN to a network.

4.1 Costs

Estimating the costs of online network management services without practical experience is difficult. The price of such services will reflect their implementation and maintenance costs, the operational expenses, and business realities (such as demand and competition). We shall focus on the operational expenses, and argue that the processing, bandwidth, and transaction costs of some of the services described above are small. We compute costs using Microsoft's Azure pricing model;[3] costs are similar for other hosted environments.

We first turn our attention to standard authentication, authorization, and directory services. Using data from [10], we computed the traffic volume for Active Directory and Kerberos, and found that it included ≈230M packets and ≈200GB of data for roughly 400 hosts and 3.5 weeks. Assuming, for simplicity, one transaction for every pair of packets, the bandwidth and transaction costs in today's prices would be less than $180/mo (plus $88/mo CPU costs).

We now consider the example of storing network information, such as network topology, configuration files, and traffic patterns. This information can be used for archiving purposes, and can also feed various network planning activities. We are particularly interested in storage since it is a rather expensive service. We again use the network studied in [10], which contains 200 backbone routers. An example of traffic patterns of interest could be the traffic volumes between all those 200 routers. For each of the $200 \times 200 = 40$K entries of the traffic matrix, we shall use 4bytes to store hourly traffic volumes. Assuming no compression of the information (despite the fact that many entries of that matrix are very small), we will spend around $1/mo to store the hourly traffic matrices for 5 years (plus some extra for meta-data information). Similarly, assuming 300K devices in our network and 1KB per device (to store basic information about a device and its connectivity) the cost of storing one topology would be ≈ $0.05/mo, (again without any compression). Similarly, we expect the cost of storing configuration files to be small. Assuming an automated process for collecting that information and uploading it to the cloud, we argue that the small storage costs make a compelling argument for storing network related information, which later can be used by other applications.

4.2 Availability

A major concern with many online services is availability [16]. Unlike traditional network management systems, where the human manager is responsible for all dependencies of a service, in the case of online services, the service itself and the network path to the cloud is beyond the control of the manager.

The current cloud offerings promise availability that is below the five nines expected by infrastructure and telecom networks. For example, Microsoft's Azure SLAs promise availability between 99.9% to 99.95% for a range of services and applications.[4] Other cloud platforms and services make similar availability promises. A 99.9% availability implies that the service is not accessible for 44min per month. If the service outages were short and evenly spread in time, then their impact on many network services (that have build-in retry mechanisms), would be small. However, long periods of outages may disrupt the operation of the network. Without hard evidence, it is difficult to assess the impact of current SLAs on cloud network management. We expect that with time, cloud operators will be able to offer more stringent availability guarantees [16].

The availability of an online service also depends on the good health of the network path between the network and the cloud, which depends on the intermediate ISPs and other (external) entities. Standard practices of negotiating appropriate SLAs and provisioning backup links, both by the network and from the cloud operator, should help in this direction. Another way employed by providers of online services to reduce the dependency on the intermediate networks is to increase their peering points [11,15]. The trend of extending the cloud closer to its customers will further improve the availability, throughput, and latency of cloud network management services.

[3] `http://www.microsoft.com/windowsazure/pricing/`

[4] `http://www.microsoft.com/windowsazure/sla/`

4.3 Other challenges

There are many other challenges and practical concerns, including information leakage and security.

We envision a cloud that will enable an avalanche of third party applications that promise to provide new services and better insights about the network. Even though it is reasonable to expect that the cloud operator protects the privacy of the data, this may not be realistic for all the 3rd parties that require detailed information about the network in order to provide extra services. Avoiding information leakage is a challenging problem ([7,14]), and should be an important concern for those wishing to utilize services on top of the cloud.

If an attacker gains access to the online management software, then he can inject arbitrary configuration state to the network. Current networks employ various techniques to fend against such attacks by, for example, restricting management access from only a well-protected sub-network, or even by creating a physical separate management plane. We should design similar analogies for cloud-based network management solutions. In the event of a security incident, the network should allow a rapid mechanism to re-create secure management channels.

5 Conclusions

This paper studies the possibility of moving management applications from the managed network to the cloud, with a focus on the question of how to deal with the increased access latency. We argue that this is not a problem for a fair fraction of management applications, either by design or due to usage model. For applications that do need to be responsive, the challenge is to be able to decompose them in a meaningful way so that the least amount of functionality still resides on the devices themselves. While we used the simple VPN scenario for ease of exposition, decomposing other latency sensitive tasks might be more convoluted. For instance, offering inter-domain routing as a service has been studied [12] while we are currently exploring ways to implement intra-domain routing as a service.

Despite this paper's narrow focus on tackling the increased latency, we are excited by the avenues that the notion of management as a service opens. For a whole host of reasons, network management has been a singular activity. Cloud-based management will change that and is bound to spur new management designs and algorithms. For instance, access to information about how a network (topology, traffic matrix, applications) has evolved over time can be used for everything from planning to debugging. Beyond this, such archives could even be used to correlate information across networks, something which is certainly not available today.

Another challenging aspect of management today is the myriad of devices, applications and interfaces. While previous efforts in the research community have proposed holistic architectures and generic abstractions to tackle this, we envision that the benefits of off-loaded network management might just nudge network owners towards homogenized software, devices and even topologies. This, of course, assumes that we can overcome the technical and business challenges posed by this new management model.

References

[1] Aggarwal, B., Bhagwan, R., and Das, T. NetPrints: Diagnosing home network misconfigurations using shared knowledge. In *NSDI* (2009), USENIX Association, pp. 349–364.

[2] Alvarez, V. A new wave of network management solutions finds success in mid-market enterprises. Tech. rep., Yankee Group, 2007.

[3] Ballani, H., and Francis, P. CONMan: a step towards network manageability. In *SIGCOMM* (2007), ACM, pp. 205–216.

[4] Benson, T., Akella, A., and Maltz, D. Unraveling the complexity of network management. In *NSDI* (2009), USENIX Association, pp. 335–348.

[5] Casado, M., Freedman, M. J., Pettit, J., Luo, J., McKeown, N., and Shenker, S. Ethane: taking control of the enterprise. In *SIGCOMM* (2007), ACM, pp. 1–12.

[6] Casado, M., Garfinkel, T., Akella, A., Freedman, M. J., Boneh, D., McKeown, N., and Shenker, S. SANE: a protection architecture for enterprise networks. In *USENIX-SS'06: Proc. of the 15th conf. on USENIX Security Symposium* (2006), USENIX Association.

[7] Dwork, C. An ad omnia approach to defining and achieving private data analysis. In *PinKDD* (2007), F. Bonchi, E. Ferrari, B. Malin, and Y. Saygin, Eds., vol. 4890 of *Lecture Notes in Computer Science*, Springer, pp. 1–13.

[8] Greenberg, A., Hjalmtysson, G., Maltz, D. A., Myers, A., Rexford, J., Xie, G., Yan, H., Zhan, J., and Zhang, H. A clean slate 4d approach to network control and management. *SIGCOMM Comput. Commun. Rev. 35* (2005), 41–54.

[9] Kandula, S., Mahajan, R., Verkaik, P., Agarwal, S., Padhye, J., and Bahl, P. Detailed diagnosis in enterprise networks. In *SIGCOMM* (2009), ACM, pp. 243–254.

[10] Karagiannis, T., and Mortier, R. Address and traffic dynamics in a large enterprise network. In *LANMAN* (2008), IEEE.

[11] Krishnan, R., Madhyastha, H. V., Srinivasan, S., Jain, S., Krishnamurthy, A., Anderson, T., and Gao, J. Moving beyond end-to-end path information to optimize CDN performance. In *IMC* (2009), ACM, pp. 190–201.

[12] Lakshminarayanan, K., Stoica, I., Shenker, S., and Rexford, J. Routing as a service. Tech. rep., Computer Science Division, University of California, Berkeley, CA, USA, 2004.

[13] McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., and Turner, J. OpenFlow: enabling innovation in campus networks. *SIGCOMM Comput. Commun. Rev. 38*, 2 (2008), 69–74.

[14] Narayanan, A., and Shmatikov, V. Robust deanonymization of large sparse datasets. In *IEEE Symposium on Security and Privacy* (2008), IEEE Computer Society, pp. 111–125.

[15] Norton, W. B. Video internet: The next wave of massive disruption to the u.s. peering ecosystem (v1.6). `http://www.drpeering.net/a/Internet_Peering_White_Papers_files/Internet%20Video%20Next%20Wave%20of%20Disruption%20v1.6.pdf`, 2008.

[16] Sripanidkulchai, K., Sahu, S., Ruan, Y., Shaikh, A., and Dorai, C. Are clouds ready for large distributed applica-

tions? In *SIGOPS Intl. Workshop on Large Scale Distributed Systems and Middleware (LADIS)* (2009), ACM.

[17] Tariq, M., Zeitoun, A., Valancius, V., Feamster, N., and Ammar, M. Answering what-if deployment and configuration questions with wise. In *SIGCOMM* (2008), ACM.

[18] Yan, H., Maltz, D. A., Ng, T. S. E., Gogineni, H., Zhang, H., and Cai, Z. Tesseract: A 4D network control plane. In *NSDI* (2007), USENIX Association.