SOFTWARE DEFINE NETWORK

Title : Study in details Cloud seeds automates IAAS using SDN and a high performance network from Juniper SDN Framework

CERTIFICATE

This is to certify that the report entitled "Study in details Cloud seeds automates IAAS using SDN and a high-performance network from Juniper SDN Framework" being submitted by is a record of Bonafede work carried out by him/her under the supervision and guidance of Prof. S. P. Vidhate in partial fulfilment of the requirement of BE Computer Engineering course of Savitribai Phule Pune University, Pune in academic year 2022- 2023.

Date: 17/5/2023

Place: Pune

Guide HOD

ACKNOWLEDGEMENT

We respect and thank Prof. S. P. Vidhate for providing us with an opportunity to do the work in college and giving us all the support and guidance. We are incredibly thankful to him for providing such nice support and guidance.

We are grateful and fortunate enough to get constant encouragement, support, and guidance from all the teaching staff of the Computer Department which helped us in the successfully.

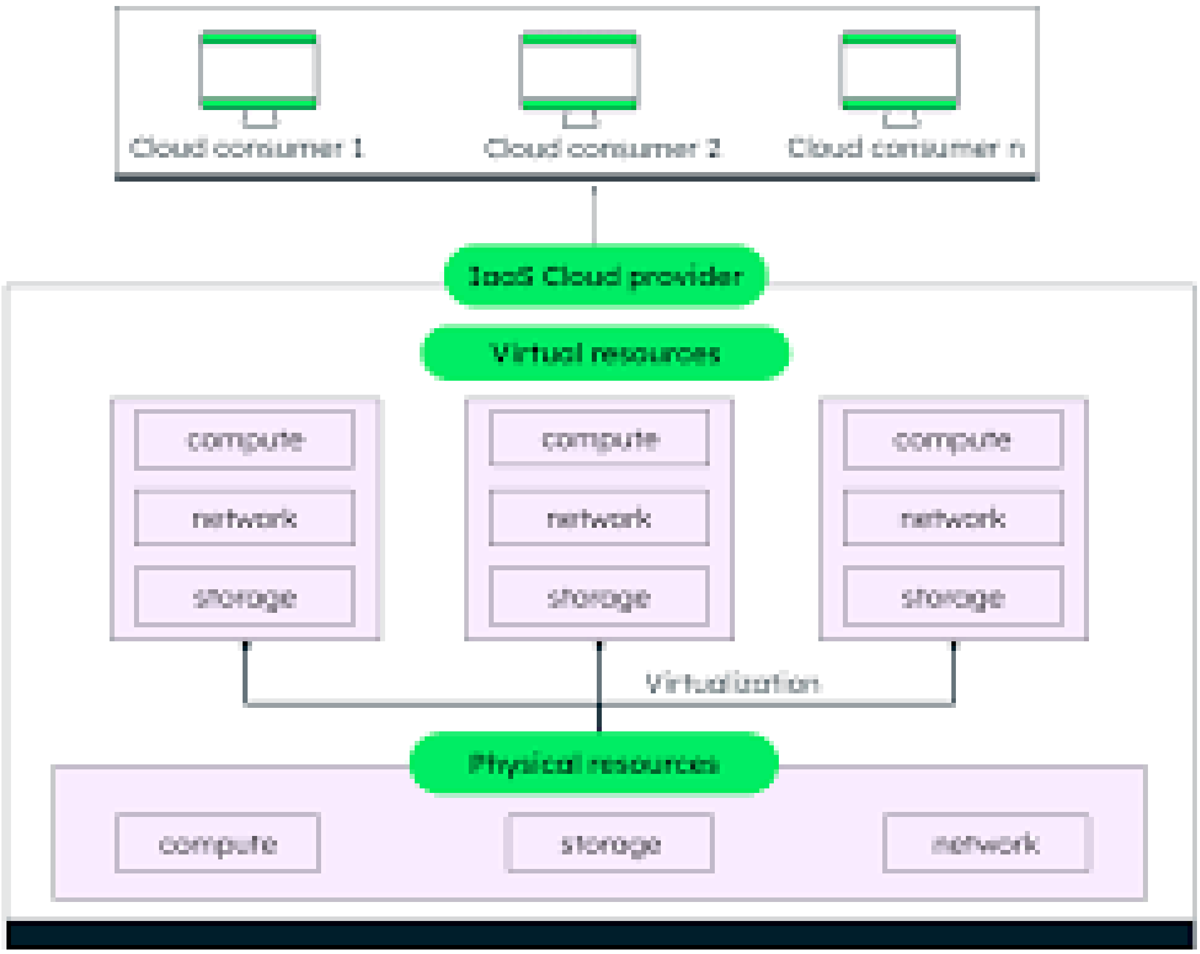
ABSTRACT

This experiment aims to investigate the use of Software-Defined Networking (SDN) and a high-performance network based on Juniper SDN Framework to automate Infrastructure as a Service (IAAS) provisioning in cloud environments. The study explores the potential benefits of leveraging SDN technologies to enhance the agility, scalability, and efficiency of IAAS deployments. The experiment involves designing and implementing a proof-of-concept system that combines SDN with Juniper SDN Framework to automate IAAS provisioning processes and evaluate its performance compared to traditional approaches.

2. **INTRODUCTION**

Infrastructure as a Service (IAAS) has revolutionized the way organizations manage their IT infrastructure by providing on-demand access to virtualized computing resources. However, manual configuration and management of IAAS platforms can be time-consuming, error-prone, and lack scalability. SDN offers a promising approach to automate IAAS provisioning by decoupling the control and data planes, enabling centralized network management and programmability. This experiment aims to explore the potential of SDN and Juniper SDN Framework in automating IAAS deployments.

Cloud Seeds is a comprehensive framework that automates Infrastructure as a Service (IAAS) provisioning using Software-Defined Networking (SDN) and the high-performance network capabilities of the Juniper SDN Framework. This section provides a detailed description of the Cloud Seeds framework, including its architecture, components, and the automation workflow involved in IAAS provisioning.



Methodology:

1. Experimental Setup:

- Setup a testbed environment comprising of physical and virtual network devices.
- Deploy Juniper SDN Framework to enable network programmability and automation.

2. IAAS Provisioning Automation:

- Develop a software application or script to orchestrate IAAS resources using SDN and Juniper SDN Framework.
- Automate the provisioning, scaling, and teardown of virtual machines, storage, and networking components.

3. Performance Evaluation:

- Measure the performance of the IAAS automation system in terms of provisioning time, resource utilization, and scalability.
- Compare the results with traditional IAAS provisioning methods to assess the benefits of SDN and Juniper SDN Framework.

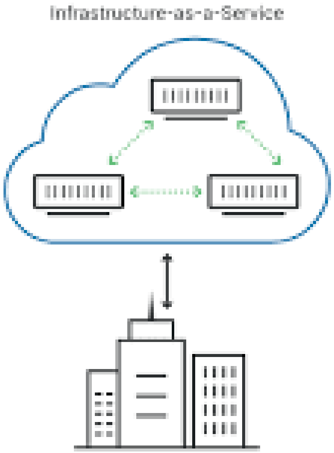
4. Analysis and Discussion:

- Analyze the experimental results and identify the advantages and limitations of the proposed automation system.
- Discuss the potential impact of SDN-based IAAS automation on cloud deployments.

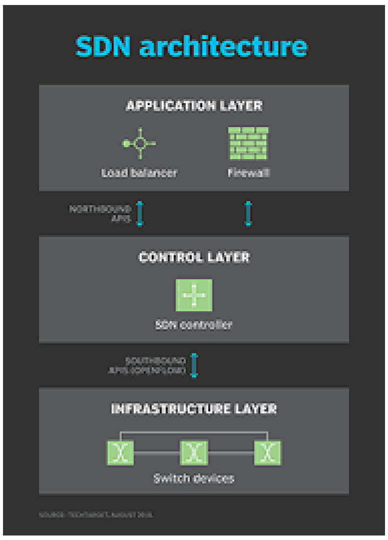
Cloud Seeds Architecture:

The Cloud Seeds architecture

a. IAAS Orchestrator: The IAAS Orchestrator is the central component responsible for managing and orchestrating the IAAS provisioning process. It communicates with the SDN controllers and the Juniper SDN Framework to automate the configuration of virtual machines (VMs), storage, and networking resources.



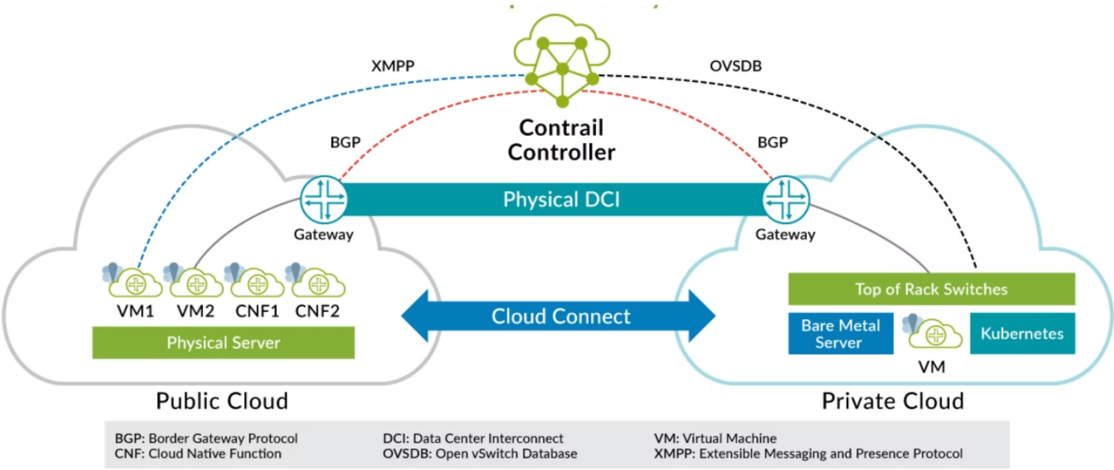
b. SDN Controllers: SDN controllers serve as the control plane in the Cloud Seeds framework. They provide a centralized management interface for configuring and controlling the network infrastructure. The controllers interact with the IAAS Orchestrator to receive provisioning requests and translate them into appropriate network configuration instructions.



c. Juniper SDN Framework: Juniper SDN Framework is a high-performance networking solution that complements the Cloud Seeds framework. It offers advanced network capabilities, including programmability, policy-based routing, and traffic optimization. Juniper SDN Framework seamlessly integrates with the Cloud Seeds system, providing a robust and efficient networking infrastructure for IAAS deployments.

One of the key benefits of the Juniper SDN framework is its ability to virtualize the network infrastructure. By abstracting the underlying physical network, it enables the creation of virtual networks that can be provisioned and configured on-demand. This virtualization brings significant flexibility and efficiency, allowing organizations to quickly adapt to changing business requirements and optimize resource utilization.

Overall, the Juniper SDN framework empowers organizations to build agile, scalable, and intelligent networks. It enables them to simplify network management, automate routine tasks, and accelerate service delivery. With its emphasis on openness, flexibility, and virtualization, the Juniper SDN framework is well-positioned to support the evolving demands of modern networking environments.



d. Virtualization Infrastructure: The virtualization infrastructure forms the foundation of the IAAS environment. It consists of virtualization hypervisors, such as VMware or KVM, which host the virtual machines. The IAAS Orchestrator interacts with the hypervisors to provision and manage the virtual resources.

IAAS ACHRITECTURE:

Infrastructure as a Service (IaaS) is a cloud computing service model that provides virtualized computing resources over the internet. In an IaaS layered architecture, the infrastructure is abstracted and provided as a service, allowing users to manage and control their computing resources without the need for physical infrastructure ownership. The IaaS layered architecture consists of the following layers:

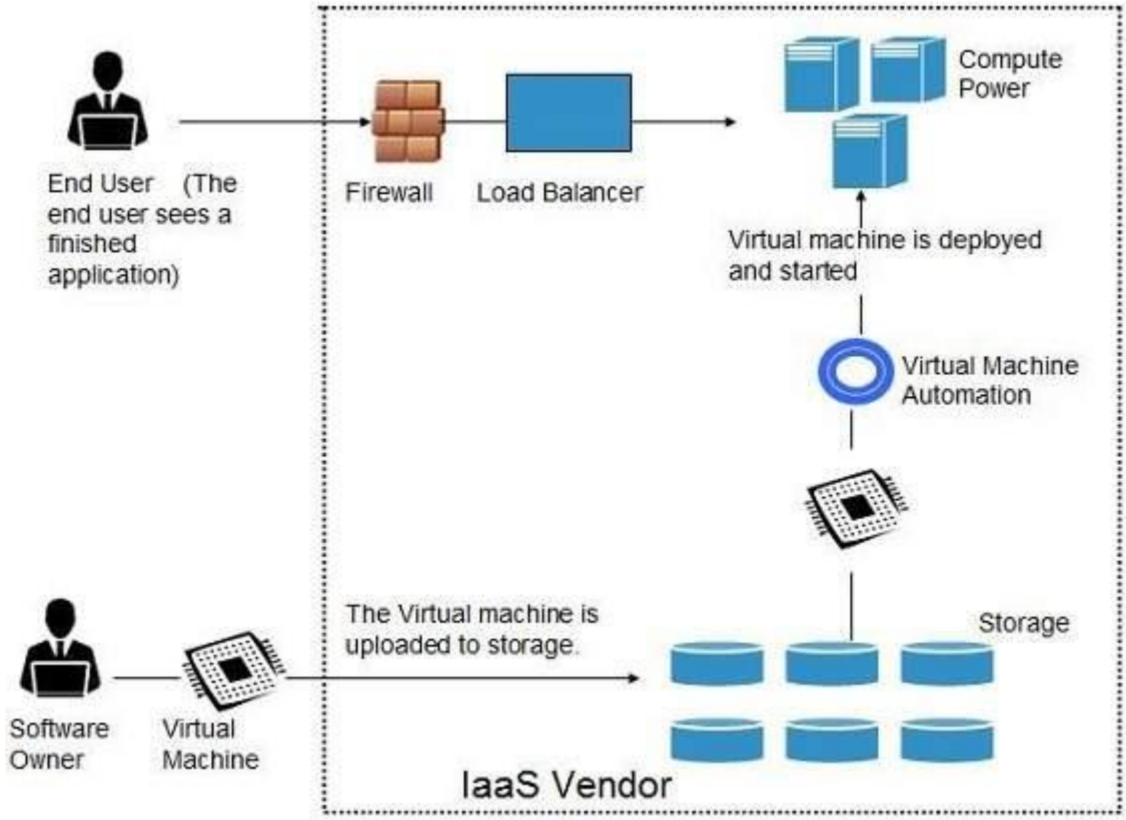
Physical Layer: At the bottom of the IaaS layered architecture is the physical layer, which comprises the actual physical infrastructure, including servers, storage devices, networking equipment, and data centers. This layer is responsible for providing the underlying hardware resources required to deliver the IaaS services.

Virtualization Layer: Above the physical layer is the virtualization layer. This layer utilizes virtualization technologies to abstract the underlying physical resources and create virtual machines (VMs) or virtual instances. Virtualization allows multiple VMs to run concurrently on a single physical server, enabling resource isolation and allocation.

Infrastructure Management Layer: The infrastructure management layer sits above the virtualization layer and is responsible for managing and provisioning the virtualized resources. It includes various components and services such as orchestration, automation, resource allocation, and monitoring tools. This layer enables users to control and manage their virtual infrastructure efficiently.

API Abstraction Layer: The API (Application Programming Interface) abstraction layer provides an interface between the infrastructure management layer and the user-facing services. It exposes APIs that allow users to interact with the IaaS platform programmatically, enabling automation and integration with other systems. The API abstraction layer facilitates provisioning, scaling, monitoring, and management of the infrastructure resources through API calls.

User-Facing Services Layer: At the top of the IaaS layered architecture is the user-facing services layer. This layer includes various services that are exposed to users for managing and utilizing the infrastructure resources. These services typically include virtual machine management, storage management, networking services, load balancers, security services, and other ancillary services. Users can provision, configure, monitor, and scale their virtual infrastructure using these services through user interfaces or APIs.



Algorithm & Implementation:

Algorithm:

1. Initialize IAAS Environment: - Set up a cloud environment using an open-source IAAS platform like OpenStack or Apache CloudStack. - Deploy virtualization technologies such as KVM or VMware to manage virtual machines.

2. Integrate Juniper SDN Framework: - Install and configure the Juniper SDN Framework, including SDN controllers and switches. - Establish communication between the IAAS platform and the SDN controller.

3. Automate Resource Provisioning: - Develop a resource allocation algorithm that determines the number of virtual machines and their configurations based on user demand. - Implement the algorithm using the Juniper SDN Framework's APIs to dynamically allocate resources.

4. Implement SDN-based Network Management: - Utilize the Juniper SDN Framework to automate network configuration tasks, including virtual network creation, subnet allocation, and security group setup. - Implement SDN applications for traffic engineering, load balancing, and QoS provisioning using Juniper SDN APIs.

5. Performance Evaluation: - Measure performance metrics such as response

time, network throughput, and resource utilization in the automated IAAS environment. - Compare these metrics with traditional manual provisioning approaches to evaluate the benefits of automation and SDN integration.

Implementation

Steps: 1. Set up the IAAS environment: - Install and configure the chosen IAAS platform (e.g., OpenStack) on dedicated servers. - Configure networking components, including switches and routers, for connectivity between the IAAS platform and SDN components.

2. Deploy Juniper SDN Framework: - Install and configure the Juniper SDN controller and switches, following the documentation provided by Juniper Networks. - Establish communication channels and APIs between the IAAS platform and the Juniper SDN controller.

3. Develop Resource Provisioning Algorithm: - Design an algorithm that takes into account user demand, resource availability, and scaling triggers to determine optimal resource allocation. - Implement the algorithm using programming languages and APIs compatible with the Juniper SDN Framework.

4. Automate Network Management: - Utilize Juniper SDN APIs to automate network configuration tasks, such as creating virtual networks, setting up subnets, and applying security policies. - Develop SDN applications using Juniper APIs to handle traffic engineering, load balancing, and QoS provisioning based on network conditions and policies.

5. Conduct Performance Evaluation: - Design a set of performance tests and benchmarks to measure response time, network throughput, and resource utilization in the automated IAAS environment. - Implement the performance evaluation framework using appropriate tools and monitoring systems. - Run the tests and analyze the collected data to evaluate the effectiveness of the automation and SDN integration in improving IAAS performance

Automation Workflow:

The automation workflow in Cloud Seeds involves the following steps:

a. IAAS Provisioning Request: A user or application submits an IAAS provisioning request to the Cloud Seeds framework. The request specifies the required VMs, storage, and networking configurations.
b. Request Processing: The IAAS Orchestrator receives the provisioning request and analyzes its requirements. It determines the appropriate network configuration based on the desired IAAS setup.

c. SDN Controller Communication: The IAAS Orchestrator communicates with the SDN controllers to translate the IAAS provisioning request into network configuration instructions. The SDN controllers leverage the programmability of Juniper SDN Framework to dynamically configure the network resources.
d. Network Configuration: The SDN controllers, using the Juniper SDN Framework, configure the networking infrastructure according to the provisioning request. This includes setting up virtual networks, defining routing policies, and implementing quality of service (QoS) measures.
e. Virtual Resource Provisioning: The IAAS Orchestrator interacts with the virtualization infrastructure, such as the hypervisors, to provision the required VMs and storage resources. It ensures that the VMs are deployed on the appropriate hosts and that the storage volumes are allocated correctly.
f. Completion and User Access: Once the provisioning process is complete, the IAAS Orchestrator provides the user or application with the necessary access credentials and connection details to access the provisioned IAAS resources.

Benefits and Advantages:

Cloud Seeds offers several benefits and advantages for IAAS provisioning:

a. Automation and Agility: By leveraging SDN and Juniper SDN Framework, Cloud Seeds automates and accelerates the IAAS provisioning process, reducing manual effort and time. It enables rapid deployment of virtual resources and facilitates the dynamic scaling of IAAS environments.

b. Enhanced Performance and Scalability: The high-performance network capabilities of Juniper SDN Framework ensure optimized data transfer, low latency, and efficient traffic routing. This results in improved performance and scalability of IAAS deployments, allowing for seamless handling of increasing workloads.

c. Centralized Management and Control: The centralized management interface provided by SDN controllers and the IAAS Orchestrator allows for unified control and monitoring of the IAAS infrastructure. It simplifies administration tasks, enhances network visibility, and enables effective resource utilization.

d. Flexibility and Customization: Cloud Seeds offers flexibility and customization options through the programmability of SDN and the advanced capabilities of Juniper SDN Framework. It enables the definition and enforcement of policies, QoS rules, and network configurations tailored to specific IAAS requirements.

Limitations and Challenges:

While Cloud Seeds provides significant benefits, some limitations and challenges should be considered:

a. Learning Curve: Adopting SDN and Juniper SDN Framework may require a learning curve for administrators and operators who are unfamiliar with these technologies. Training and expertise are essential to effectively utilize the framework's capabilities.

b. Integration Complexity: Integrating the Cloud Seeds framework with existing IAAS deployments or legacy systems may pose challenges. Ensuring compatibility, data migration, and coexistence with conventional networking components may require careful planning and implementation.

c. Security and Compliance: As with any IAAS environment, ensuring robust security measures and compliance with industry standards and regulations is crucial. Cloud Seeds should incorporate appropriate security controls, encryption, access management, and monitoring mechanisms to protect IAAS resources and user data.

d. Scalability and Performance Testing: It is important to conduct thorough testing and performance evaluations to assess the scalability and performance of the Cloud Seeds framework under various workload scenarios. This ensures that the system can handle increasing demands without compromising performance or stability.

Conclusion:

This study aims to investigate the automation of IAAS provisioning using SDN and the Juniper SDN Framework. By combining the agility and programmability of SDN with Juniper's robust network infrastructure, the experiment seeks to demonstrate the potential benefits of automating IAAS deployments. The results of this experiment will contribute to the growing body of knowledge on SDN-enabled cloud automation and provide insights into the feasibility and performance of the proposed approach.

References:

[1]AndreasMetzgerandClarissaCassalesMarquezan.“FutureInternetApps:TheNext WaveofAdaptiveService-OrientedSystems?”In:TowardsaService-BasedInternet:4th EuropeanConference,ServiceWave2011,Poznan,Poland,October26-28,2011.Proceedings.Ed.byWitoldAbramowiczetal.Berlin,Heidelberg:SpringerBerlinHeidelberg, 2011,pp.230–241.ISBN:978-3-642-24755-2.DOI:10.1007/978-3-642-247552_22.URL:http://dx.doi.org/10.1007/978-3-642-24755-2_22.

[2]TheophilusBenson,AdityaAkella,andDavidMaltz.“UnravelingtheComplexity ofNetworkManagement”.In:Proceedingsofthe6thUSENIXSymposiumonNetworked SystemsDesignandImplementation.NSDI’09.Boston,Massachusetts:USENIXAssociation,2009,pp.335– 348.URL:http://dl.acm.org/citation.cfm?id= 1558977.1559000.

[3]NickMcKeown.“Softwaredefinednetworking”.In:vol.17.2.2009,pp.30–32.

[4]HyojoonKimandN.Feamster.“Improvingnetworkmanagementwithsoftwaredefinednetworking”.In:C ommunicationsMagazine,IEEE51.2(2013),pp.114–119.ISSN: 0163-6804.DOI:10.1109/MCOM.2013.6461195.

[5]R.Mijumbietal.“NetworkFunctionVirtualization:State-ofthe-ArtandResearch Challenges”.In:vol.18.1.2016,pp.236–262.DOI:10.1109/COMST.2015.2477041.

[6]B.Hanetal.“Networkfunctionvirtualization:Challengesandopportunitiesforinnovations”.In:vol.53.2.2 015,pp.90–97.DOI:10.1109/MCOM.2015.7045396.

[7]D.Kreutzetal.“SoftwareDefinedNetworking:AComprehensiveSurvey”.In:ProceedingsoftheIEEE103.1(2015),pp.14– 76.ISSN:0018-9219.DOI:10.1109/JPROC. 2014.2371999.

[8]InternetWorldStats."WorldInternetUsageandPopulationStatistics".2017.URL:http: //www.internetworldstats.com/stats.htm.

[9]AndrewT.Campbelletal.“ASurveyofProgrammableNetworks”.In:SIGCOMM Comput.Commun.Rev.29.2(Apr.1999),pp.7–23.ISSN:0146-4833.DOI:10.1145/ 505733.505735.URL:http://doi.acm.org/10.1145/505733.505735.

[10]J.Biswasetal.“TheIEEEP1520standardsinitiativeforprogrammablenetworkinterfaces”.In:IEEEComm unicationsMagazine36.10(1998),pp.64–70.ISSN:0163-6804. DOI:10.1109/35.722138.

[11]D.L.Tennenhouseetal.“Asurveyofactivenetworkresearch”.In:IEEECommunicationsMagazine35.1(19 97),pp.80–86.ISSN:0163-6804.DOI:10.1109/35.568214 SIGCOMMComput.Commun.Rev.38.2(Mar.2008),pp.69–74.ISSN:0146-4833.DOI: 10.1145/1355734.1355746.URL:http://doi.acm.org/10.1145/ 1355734.1355746