Autonomous Network Management Using Cooperative Learning for Network-Wide Load Balancing in Heterogeneous Networks

Conference Paper · January 2009

DOI: 10.1109/GLOCOM.2008.ECP.489 · Source: IEEE Xplore

Autonomous Network Management Using Cooperative Learning for Network-Wide Load Balancing in Heterogeneous Networks

Minsoo Lee, Xiaohui Ye, Dan Marconett, Samuel Johnson, Rao Vemuri and S. J. Ben Yoo
Department of Electrical and Computer Engineering,
University of California Davis,
Davis, CA 95616,
E-mail: {msolee, xye, dmarconett, samjohnson, rvemuri, sbyoo}@ucdavis.edu

Abstract— **Traditional hop-by-hop dynamic routing makes inefficient use of network resources as it forwards packets along already congested shortest paths while uncongested longer paths may be underutilized. To maintain network-wide load balancing, we propose Autonomous Network management with Team learning based Self-configuration (ANTS) which attempts to manage a feasible route for traffic flow with QoS constraints in heterogeneous networks. To enable cognitive intelligence for network-wide load balancing, we implement a cross-layer mechanism in which learning agents in middleware layer can monitor the queue sizes of MAC layer, thereby allowing for the discovery of optimal routes. We present OPNET simulation results illustrating that, in comparison to original OSPF and AODV (2.18 Mbits/s with 46.46% packet loss rate), ANTS dramatically achieves a higher packet delivery (9.57 Mbits/s with 0.53% packet loss rate).**

Keywords-cognitive networks, load balancing, reinforcement learning, wireless networks

I. INTRODUCTION

Traditional hop-by-hop dynamic routing like Open Shortest Path First (OSPF) [1] forwards packets along the shortest path based purely on destination address in the packet header. Such an approach is sufficient for best-effort data transport, but makes inefficient use of network resources as it can forward packets along already congested shortest paths, not allowing longer but less-congested paths to be utilized [2]. We make the case that maintaining a network-wide load balance during the route selection process is critical for improving network-wide throughput and utilization. We assert that team cognitive intelligence during the route selection process plays a decisive role in maintaining the network-wide load balancing. Specifically, our research is concerned with the following questions:

1. How should the cognitive intelligence of learning agents be used for network-wide load balancing?

2. How can reinforcement learning agents cooperate to achieve network-wide load balancing?

3. What policies should be enforced by an agent with respect to the learning results of other agents, and exchanged by data communications?

Toward the above stated concerns, and the goal of maintaining network-wide load balance, we propose Autonomous Network management with Team learning based Self-configuration (ANTS) which attempts to find a feasible route for a traffic flow with QoS constraints of a given network environment. We implement ANTS in a cross-layer context, demonstrating that it overcomes limitations of current network management schemes in heterogeneous wireless networks by allowing network agents to *observe, analyze,* and *act* [3] in order to optimize their performance. Our approach is to modify the routing strategy with team-based learning, to ensure that the Service Level Agreements (SLA's) and packet delivery ratio can be achieved at desired levels while minimizing management overhead. We describe the advantages of our approach, summarized in Table I, with respect to a qualitative analysis of the challenges faced in heterogeneous networks. We present OPNET [4] simulation results showing that ANTS successfully improves the scalability and resource utilization of the OSPF and AODV routing protocols in a heterogeneous network environment.

II. RELATED WORK

Many previous works on network management have been devoted to providing QoS guarantees and preventing network congestions. Network service providers are responding to QoS guarantee demands with QoS-guaranteed-VPN (QVPN), which utilizes technologies such as Multi-Protocol Label Switching (MPLS). However, a major challenge is how to map each QVPN into a physical network path such that as many QVPNs as possible can be supported on its network infrastructure [2]. Solutions such as QoS routing mechanisms [5] [6] attempt to find a route for a traffic flow with QoS constraints. However, the rapid growth in various real-time applications in heterogeneous networks has made it indispensable to consider the impact of end-to-end delay traffic requirements with resource provisioning decisions on the scale of the entire network. Recent solutions with QVPN [2] address the network-wide load balancing but these account only for the maximum bandwidth requirements, not for actual loads. Current network management schemes can be

978-1-4244-2324-8/08/$25.00 © 2008 IEEE.
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2008 proceedings.

TABLE I. OUR AUTONOMOUS NETWORK MANAGEMENT WITH TEAM LEARNING BASED SELF-CONFIGURATION (ANTS)

Key Challenges	Key Technologies	Our Approach
How should the cognitive intelligence of learning agents be used for network-wide load balancing?	– Autonomic Network Management	– Implement new functions in network management plane – NOT modifying the existing protocols
	– Route Management – Network Monitoring	– Discover optimal routes based on network observation – Reconfigure routing protocols to discover more-optimal paths – A Learning agent in middleware layer monitors the queue size of MAC layer as the key context of cognitive intelligence for load balancing.
How can reinforcement learning agents be cooperative for network-wide load balancing?	– Network-wide Load Balancing	– Cognizant of underlying technology (wired, wireless)
	- Cooperative Machine Learning	– Use appropriate machine learning technology to optimize in a multi-agent environment – The Q-learning agents in wired networks and wireless networks can exchange cognitive intelligence
What policies should be enforced by an agent with respect to the learning results of other agents, and exchanged by data communications?	– Machine Learning for Performance Enhancement – Cross-Layer Approach	– Use cross-layer information to optimize decisions of agents – Cross-layer resource management from network layer to application layer

significantly enhanced with the network-wide knowledge using cooperative learning agents in a cross-layer approach.

Agents having a limited access to relevant network information run the risk of failing in solving a given learning task. The risk may be reduced by letting agents exchange relevant context including the network environment in which they are embedded, strategies and knowledge of other agents, the dependencies among different actions and the reward of other agents [7]. Ming Tan [8] showed that a cooperative learning agent will outperform independent agents. Using cooperative learning in a cross-layer approach, we implement a communication method that can let the intelligent agents in wired networks and wireless networks cooperate to achieve network-wide load balancing. For network-wide load balancing in wireless mesh networks, gateway load balancing in incorporated into AODV-ST [9]. However it only focuses on the multi-radio relay nodes that comprise the *infrastructure mesh network*; the end-devices do not participate in the packet relay. This paper primarily focuses on *client mesh network* wherein end-devices participate in packet forwarding.

III. GLOBAL OPTIMAL LINK UTILIZATION BY LOAD BALANCING WITH COOPERATIVE LEARNING

A. Q-learning for Autonomous Network Management

Our cooperative learning algorithm discovers a routing policy that minimizes the possibility of congestion along popular routes. To achieve this, the algorithm experiments with different routing policies and gathering statistics about which actions minimize total packet loss. The task of adaptive routing here is to answer the question: *To which adjacent node should the current node send its packet, so that the packet can reach its destination as quickly as possible minimizing packet loss?* Here, the "state" of the reinforcement learning system is the *current node* that the packet is in, and the "action" is to choose a neighbor node as the next hop.

Our learning scheme is based on a *Q*-learning approach [10], which uses the $Q(s,a)$ function to measure the expected long-term penalty obtained by taking action a from state s and following an optimal policy thereafter. When an agent selects action a at the current state s, the agent will receive an immediate reward r, and the system state changes to s'. The agent then uses this reward and the expected long-term reward to update the Q-values, which in turn influences future action selection. Its simplest form, one-step Q-learning, is defined as:

$$Q(s,a) = (1-\alpha)Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a')]$$

where $\alpha \in [0,1]$ is the learning rate, which models the rate of the updating of Q-values. A high α-value causes the system to place more emphasis on immediate environmental feedback (reward) where a low α-value emphasizes previous feedback. The parameter γ represents the discount factor which weights the importance of expected future rewards. As a model-free Reinforcement Learning technique, Q-learning requires no knowledge about the underlying reward or transition mechanism; thus it is applicable to the problem of learning routing strategy in ad hoc networks, where explicit state-space mapping can become computationally cumbersome.

B. Cross-layer design for heterogeneous networks

To realize desired QoS guarantees it would be critical to consider the change of user demands in the application layer. For consistent QoS support, the MAC layer could provide the essential context. The MAC layer could provide an indication of the network congestion level and achievable data rates; these could be used to determine whether the lower layer capability can meet the upper layer requirement. In the proposed cognitive networking, the MAC layer in the wired node provides the queue size which represents the link congestion of the node. A cognitive agent manages a Q-value for each route path and uses it to make decisions to form a routing path for a traffic demand. Q-value estimates the quality of the path based on congestion information from MAC layers in the wired nodes through the cross-layer approach. These Q-values are updated every second.

Fig. 1 shows the cross-layer architecture for the proposed cognitive network management framework. One of the main advantages of cross-layer design is to make protocols aware of the current network state in a localized but distributed fashion. By introducing MAC, network and application layer contexts to the network management agents, the higher-level processes of the middleware is improved. It allows ANTS to exploit broader knowledge of network state, and improves overall system performance. Other proposals for implementation of cross-layer information can be categorized in three main groups [11]: (a) direct communication between layers, (b) a shared database across the layers, and (c) completely new abstractions. Specifically, we present the cross-layer interactions among layers by a shared network status module, supporting vertical communications among the layers by acting as a repository for information collected by network protocols.

C. *Proof of concept: ANTS in Wired Networks*

The following example involving ANTS demonstrates that cognitive intelligence of cooperative reinforcement learning agents significantly improves the performance of dynamically changing networks. Fig. 2 (a) shows the OPNET simulation topology of the experiment. We investigate the routing dynamics of a traffic flow from "source" to "destination." There are 6 core routers (from node_1 to node_6), 4 ingress nodes (ingress_1 to ingress_4), and 3 pairs of source and destination nodes. In each core router, a "penalty" is set based on its queue size. Each node does load balancing based on the penalty values of core routers. In the simulation, three traffic demands of 80 Mb/s are generated:

- source_1 → destination_3 (traffic #1),
- source_2 → destination_4 (traffic #2),
- source_3 → destination_1 (traffic #3).

The link capacity of all links is 100Mb. Fig. 2 (b) and (c) show the successful packet delivery of the three traffic sources with OSPF and ANTS. In the OSPF simulation scenario, the traffic flow #1 arrives (Poisson process with the average rate 80 Mb/s) at the 40-second. For the traffic flow #1, ingress_1 router finds a shortest path "ingress_1" → "node_1" → "node_2"→"ingress_3". At the 90-second, the traffic flow #2 arrives and ingress_3 router chooses the shortest path ("ingress_3" → "node_2" → "node_1" → "ingress_1"; the reverse path for traffic flow #1) for traffic flow#2. Since the ingress routers always choose the shortest paths regardless of the network dynamics, there is huge packet loss of traffic #3. At the 140-second, traffic flow #3 arrives and the packet loss becomes even worse (Fig. 2 (b)) since ingress nodes are unaware of the congestion. The four ingress nodes maintain

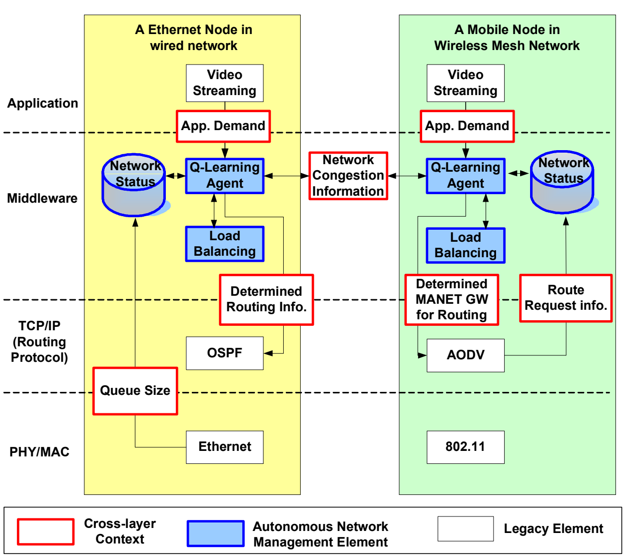


Fig. 1. Cross-layer design in Cognitive Networking

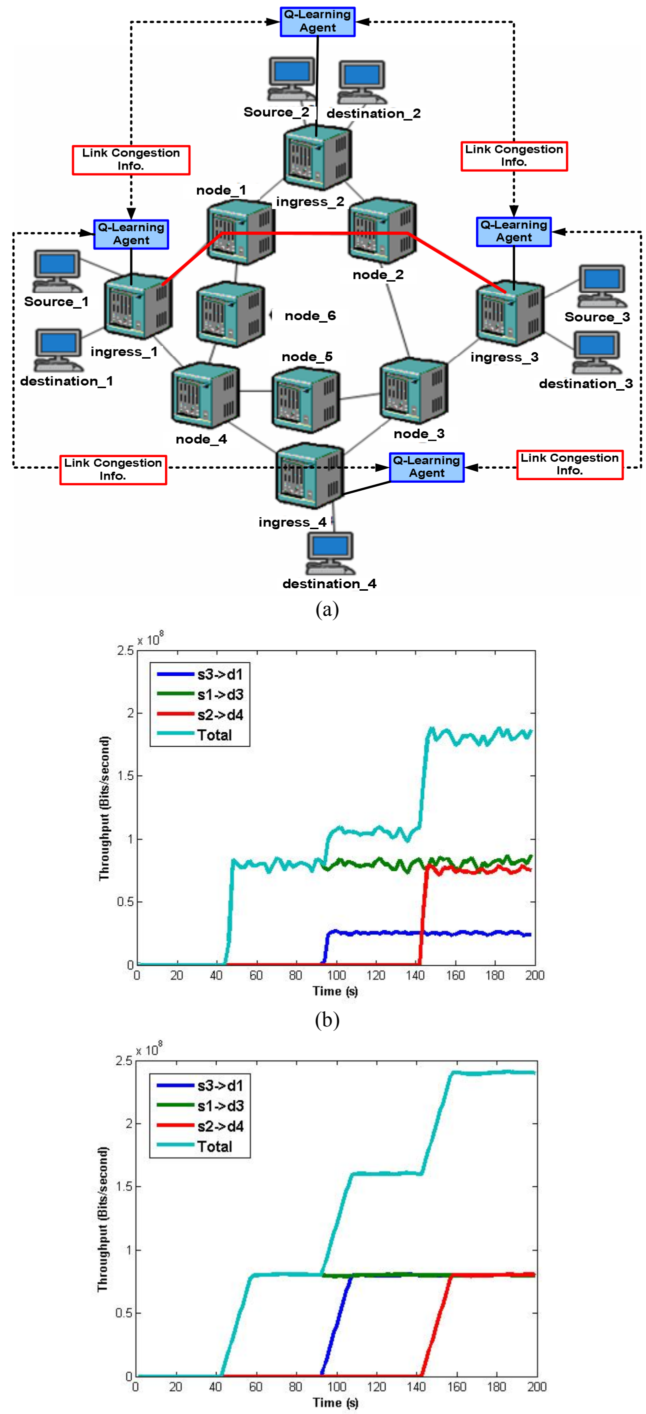


Fig. 2. OPNET simulations in wired networks for testing the roles of cognitive intelligence of cooperative reinforcement learning agents (a) Simulation topology. Packet delivery of 3 traffics (b) OSPF (c) ANTS

network-wide load balance with cognitive intelligence during the route selection process (Fig. 2 (c)). When all three traffics are generated, the average successful packet delivery with OSPF is 175 Mbit/s and the average of successful packet delivery with ANTS is 240 Mbit/s.

IV. ANTS IN HETEROGENEOUS NETOWKRS

A. *Autonomous Network Management flow*

The true benefit of autonomous networking becomes more apparent in heterogeneous environments where many different networking technologies (wireless, wireline, optical, etc) coexist. The agents, which reside in the wireless and wireline domains as well as the gateway, will optimize adaptively in collaboration with other domains Wireline routers update their queue size periodically to the local learning agents sitting in the wireline network, as shown in the implementation pseudo code in Fig. 3 (a). In a hierarchical structure of learning agents, local learning agents will also exchange information with higher layer agents once a while. Gateway nodes serve as the broker between wireline network and wireless network. Learning agents residing in gateway nodes collect router statistics from wireline network higher layer agents and distribute them to wireless learning agent if necessary. For a particular ongoing demand across wireless and wireline networks, there may be multiple possible paths through

```
Get Node_ID_current by using current Process_ID_current
FOR each node index
   Get Node_ID_index by using node index
   IF Node_ID_index = Node_ID_current
              // Node identified
              FOR each queue index
                 Get Process_ID_index by using queue index
                 IF Process_ID_index = Process_ID_current
                      // Process identified
                       Get current queue statistics
            Update corresponding entry in the learning agent
                 ENDIF
              END
   ENDIF
END
```

(a)

```
// Q-value update
FOR each possible path (at each gateway)
      Get queue size from each router in the path
      Generate Q-value for the path by accumulating queue sizes
END
// Path selection
Select the path with the smallest Q-value (Q_candidate) as candidate
IF Q_current_path – Q_candidate > Q_threshold
      Switch to the candidate path
ENDIF
```

(b)

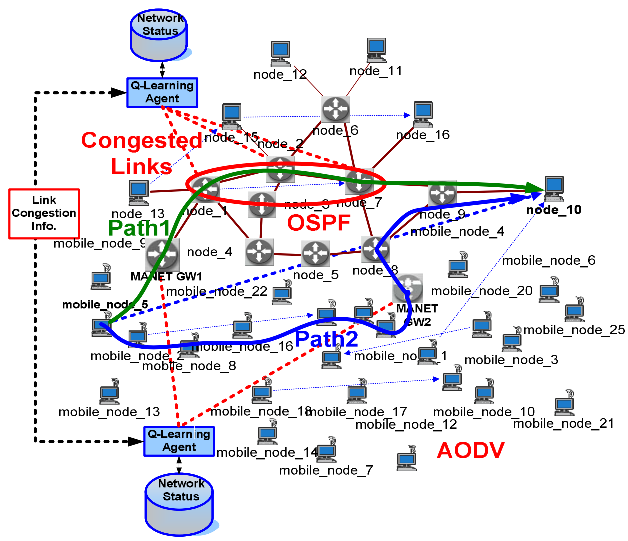
```
FOR each possible path (at each gateway)
      Get queue size from each router in the path
      Generate Q-value for the path by accumulating queue sizes
      IF Q-value > Q_threshold
              Send Route ERROR
      ENDIF
END
```

(c)

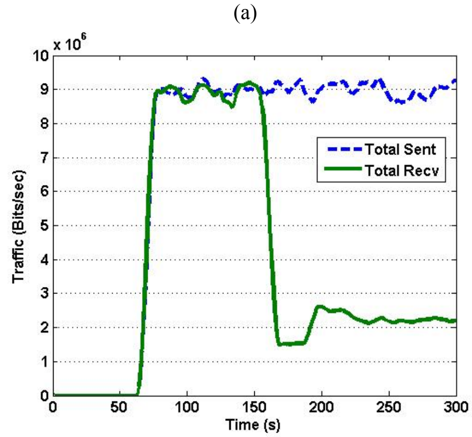
Fig. 3. Implementation pseudo codes for load balancing using Team based Learning (a) update of queue size of each router, (b) Q-value update and Path selection, and (c) Q-value update and usability check

different gateways. Thus each gateway agent will generate a Q-value for the path across itself. Since the Q-values relate to queue sizes and the number of hops in the wireline network, a higher Q-value indicates that the path is undesirable.

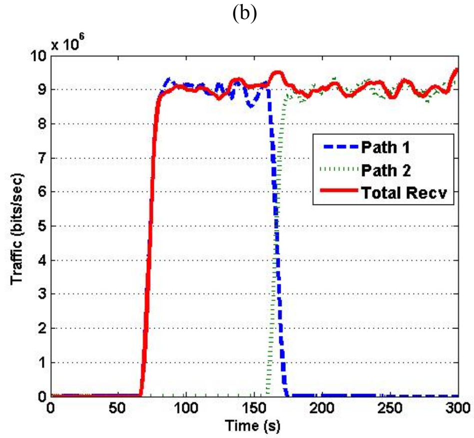
There are two different strategies to utilize Q-values generated at gateway agents: proactive and reactive. In the



(a)



(b)



(c)

Fig. 4. Cognitive Networking OPNET simulations in heterogeneous networks (a) topology, (b) with OSPF (wireline) and AODV (wireless), and (c) with team-based cognitive networking.

978-1-4244-2324-8/08/$25.00 © 2008 IEEE.
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2008 proceedings.

proactive manner, the Q-values for each possible path are periodically distributed to learning agents in the wireless network. The wireless learning agent will finally be responsible for path selection according to the multiple Q-values for different paths. In our example, each wireless end node will have its own learning agents. Fig. 3 (b) shows pseudo codes for Q-value updates and path selections. If the Q-value of the path currently using is much larger than the smallest Q-value among multiple paths, the learning agent of the wireless end node will instruct the routing function to switch to the path with the smallest Q-value since the current path may experience congestions at the time. Here a threshold is used to gauge Q-value difference. The threshold value quantifies the extent to which an agent will tolerate a sub-optimal path before switching to a better one. A lower threshold value causes the agent to switch paths more opportunistically which, if set too low, may lead the agent to oscillate between paths. Higher threshold values correspond to a higher tolerance of a sub-optimal path. In a reactive manner, a gateway agent will determine if the path across itself is congested or not for one particular demand. If it is congested, the agent will instruct the gateway to send out Route ERROR back to the corresponding wireless learning agent residing in the end node indicating the path is not usable currently. And then the wireless learning agent will help the end node choose a path from other possible candidates. When the congestion diminishes, the gateway will stop sending Route ERROR and the path will be re-established. Fig. 3 (c) shows the pseudo codes of this reactive function.

B. OPNET Simulations and Performance Analysis

The performance of our network system has been evaluated with the OPNET simulation tool [4]. We modeled a network in a flat terrain of 3km × 3km shown in Fig. 4 (a) with two regions: region 1 is comprised of Ethernet wirelined network and region 2 consists of 25 wireless nodes and 2 MANET gateways in a mobile mutihop scenario. The mobile nodes were equipped with an IEEE 802.11b wireless network interface with a data rate of 11 Mb/s. The mobile nodes are assigned a maximum speed of 15 m/s changing its location within the network based on the random waypoint model. In order to calculate the impact of high mobility on the protocol overhead, the mobile nodes are constantly moving.

Baseline performance was measured in terms of how well OSPF (wirelined region) interact with AODV (wireless region) to facilitate load balancing, where potential paths are wirelined and wireless, for a given source destination pair (“mobile_node_5”→“node_10”). In particular, we introduced background traffics at t=150s (between “node_13”, “node_15”, “node_16” and “node_7”) to purposely congest the wirelined path to see if indeed the two routing protocols across the two domains would attempt to adaptively achieve traffic engineering. As Fig. 4(b) indicates, this did not happen in the conventional OSPF and AODV.

On the other hand, Fig. 4. (c) illustrates the success in introducing a team-based *Q*-learning scheme where knowledge of network state is shared between intelligent network management agents within each region. Here a more optimal load balancing scheme, which learned the relative benefits of a multipath solution to network congestion, resulted in nearly 5 times higher throughput during the congestion (t>150s) : 9.57 Mbit/s (0.53% packet loss rate) vs. 2.18 Mbit/s (46.46% packet loss rate).

V. Conclusions and Future Research

In this paper, we described a framework of autonomous network management using cooperative learning, ANTS, for network-wide load balancing. ANTS has been proposed to improve the performance of OSPF and AODV, through the use of network state observation in large heterogeneous networks. We also presented the experimental results of ANTS using OPNET. The performance results confirm that in comparison to original OSPF and AODV (2.18 Mbits/s with 46.46% packet loss rate), ANTS dramatically achieves a higher packet delivery (9.57 Mbits/s with 0.53% packet loss rate). Several open research issues remain regarding autonomous network management using cooperative learning. The merit of the Q-Learning technique has been shown by increased performance, however, it remains to be seen whether other existing or modified machine learning techniques, such as Bayesian Learning Networks or Temporal Difference Learning, will lead to further increases in performance, in both the wirelined and wireless domains. There is also a need to investigate optimization accuracy and the process of reward value assignment in the Q-value computation, in addition to the selecting of more efficient parameters for self-configuration in various network elements.

References

[1] J. Moy, "OSPF Version 2," RFC 2328, Apr. 1998.

[2] K. Gopalan, K. Gopalan, C. Tzi-cker, and L. Yow-Jian, "Load balancing routing with bandwidth-delay guarantees," Communications Magazine, IEEE, vol. 42, pp. 108-113, 2004.

[3] U. Li Yin, S., Palmer, J., Katz, R., Agha, G., "AutoLoop: automated action selection in the "observe-analyze-act" loop for storage systems. ," in Sixth IEEE International Workshop on Policies for Distributed Systems and Networks, 2005.

[4] "http://www.opnet.com."

[5] C. Shigang, C. Shigang, and K. Nahrstedt, "On finding multi-constrained paths," in Communications, ICC 98. Conference Record.1998 IEEE International Conference on, 1998.

[6] R. A. Guerin, R. A. Guerin, A. Orda, and D. Williams, "QoS routing mechanisms and OSPF extensions," in Global Telecommunications Conference, 1997. GLOBECOM '97., IEEE, 1997, pp. 1903-1908 vol.3.

[7] G. Weiss, Multiagent Systems a Modern Approach to Distributed Artificial Intelligence: The MIT Press, 2000.

[8] M. Tan, "Multi-Agent Reinforcement Learning: Independent vs. Cooperative Learning," in Readings in Agents, M. N. H. a. M. P. Singh, Ed.: Morgan Kaufmann, 1997, pp. 487-494.

[9] M. M. B. Krishna Ramachandran, Girish Chandranmenon, Scott Miller, Elizabeth Belding-Royer and Kevin Almeroth, "On the Design and Implementation of Infrastructure Mesh Networks," in WiMesh 2005.

[10] C. W. a. P. Dayan., "Machine Learning," 1992, pp. 279-292.

[11] V. Srivastava and M. Motani, "Cross-layer design: a survey and the road ahead," Communications Magazine, IEEE, vol. 43, pp. 112-119, 2005.