A Markov Decision Process based Flow Assignment Framework for Heterogeneous Network Access *

Jatinder Pal Singh (jatinder.singh@telekom.de)
Deutsche Telekom Laboratories, 5050 El Camino Real, Los Altos, CA 94022, USA

Tansu Alpcan (tansu.alpcan@telekom.de)
Deutsche Telekom Laboratories, Ernst-Reuter Platz 7, 10587 Berlin, Germany

Piyush Agrawal (piyushag@stanford.edu)
Department of Electrical Engineering, Stanford University, Stanford, CA 94305, USA

Varun Sharma
Lehman Brothers Inc., India

Abstract. We consider a scenario where devices with multiple networking capabilities access networks with heterogeneous characteristics. In such a setting, we address the problem of efficient utilization of multiple access networks (wireless and/or wireline) by devices via optimal assignment of traffic flows with given utilities to different networks. We develop and analyze a device *middleware* functionality that monitors network characteristics and employs a Markov Decision Process (MDP) based control scheme that in conjunction with stochastic characterization of the available bit rate and delay of the networks generates an optimal policy for allocation of flows to different networks. The optimal policy maximizes, under available bit rate and delay constraints on the access networks, a discounted reward which is a function of the flow utilities. The flow assignment policy is periodically updated and is consulted by the flows to dynamically perform network selection during their lifetimes. We perform measurement tests to collect traces of available bit rate and delay characteristics on Ethernet and WLAN networks on a work day in a corporate work environment. We implement our flow assignment framework in ns-2 and simulate the system performance for a set of elastic video-like flows using the collected traces. We demonstrate that the MDP based flow assignment policy leads to significant enhancement in the QoS provisioning (higher rate allocation, lower packet delays and packet loss rates) for the flows and better access network utilization, as compared to policies (dynamic or static) that allocate flows to different networks using greedy approaches or heuristics like average available bit rate on the networks.

Keywords: Heterogeneous access networks, flow assignment, Markov Decision Process, simulative performance evaluation

* Research Supported by Deutsche Telekom AG. An earlier version of the work appeared in the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, June 18-21, 2007, Helsinki, Finland.

1. Introduction

Several networking technologies have evolved and become popular over the past few decades. Ethernet, DSL, cellular wireless networks, and IEEE 802.11 based wireless local area networks have become widely deployed and increasingly accessible. Existing networks tend to be heterogeneous in their attributes such as the supporting infrastructure, protocols, signaling mechanisms, offered data rates, etc. With the realization that several technologies will continue to coexist and there will be no clear winner, the drive towards convergence of networks is gaining momentum. Integration of heterogeneous access networks is part of the 4G network design [23]. IEEE 802.21 [1] is delineating a framework to enable handovers and interoperability between heterogeneous wireless and wireline networks. The IP Multimedia Subsystems (IMS) [8] has defined an overlay architecture for providing multimedia services on top of heterogeneous networks.

It is today commonplace to have electronic devices with multiple networking capabilities. Personal computers and laptops typically come equipped with a built-in WLAN card, a PCMCIA slot, and an Ethernet port. PDAs with WLAN and GPRS connectivity are becoming popular. As a multitude of bandwidth demanding applications such as IPTV and Internet Video run on devices, a single network may often not be sufficient to meet the requirements of the applications. Several interesting scenarios may be envisioned. Imagine a user in a corporate setting participating in a video conference call via her device having both Ethernet and IEEE 802.11g connectivity. While engaged in the conference proceedings, the user is uploading content on a remote server for the participants to access, and at the same time needs to retrieve some files from the server. Several traffic flows are hence created by the device which dynamically monitors the networks at its disposal. The device then routes the flows via these networks and dynamically reassigns them to different networks based on the varying network characteristics like available bite rate (ABR) and delay.

While the distribution of traffic flows amongst different networks can enable better network utilization than single network use at a time, the variation in network characteristics like ABR and delay makes the problem of flow assignment challenging. Especially when the access networks include wireless links, the network characteristics variations require robust modeling techniques and stochastic tools. In this work, we address the problem of optimal allocation of flows on a device onto multiple networks with heterogeneous characteristics. We approximate the ABR and delays of the networks to represent the states of a Markovian system. We then develop and analyze a *middleware* functionality that monitors the network characteristics and uses a Markov Decision Process (MDP) [6] based control scheme to suggest a network to which a flow with given utility should be assigned. The MDP selects a network that maximizes a discounted reward which is represented as a function of flow utility and the impact of the flow assignment on the system. The flow utility in turn depends on the ABR and delay offered by a network to the flow. The MDP based flow assignment policy is updated periodically by the middleware and is dynamically consulted by the flows during their lifetimes to select the suggested networks. We implement the flow assignment framework in ns-2 [2] and collect ABR and delay traces for Ethernet and WLAN networks in a real-world setting. We then evaluate the performance of high bit rate elastic video-like flows using MDP based flow assignment against dynamic and static flow assignment policies, and demonstrate that MDP based flow assignment scheme results in significantly better QoS provisioning for the flows in terms of lower packet delays and packet loss rates.

The rest of this paper is organized as follows: We discuss related work in Section 2. We present the system model and analytical framework for flow assignment in Section 3. In Section 4, we describe results from measurement tests conducted for heterogeneous access networks. The performance evaluation of the flow assignment framework is presented in Section 5. We conclude the work in Section 6.

2. Related Work

In general, the problem of efficient utilization of multiple networks via suitable allocation of traffic flows has been explored in different settings and from different perspectives. A game theoretic framework for bandwidth allocation for elastic services in networks with fixed capacities has been addressed in [24, 4, 3]. Our work on the other hand is motivated by the practically observed and varying characteristics of networks that are widely deployed today. Packet scheduling for utilization of multiple networks has been investigated in [7]. The opportunistic scheduling of packets has the drawback of needing a packet level scheduler and frequent packet reordering at the receiver. In our work, we thus focus on flow based scheduling for heterogeneous networks. A solution for addressing the handoff, network selection, and autonomic computation for integration of heterogeneous wireless networks has been presented in [23]. The work, however, does not address efficient simultaneous use of heterogeneous networks and does not consider wireline settings. Similarly, the work [9] focuses on selection techniques for users to get connected to the most suitable network in terms of user defined QoS criteria, and does not address a multi-homed device scenario. In [25], the authors have explored design of a network comprising wide area and local area technologies where user devices select among the two technologies in a greedy fashion so as to maximize a utility function based on wireless link quality, network congestion, etc. The work does not address simultaneous use of the two technologies by the users. Recently, a cost price mechanism that enables a mobile device to split its traffic amongst several IEEE 802.11 access points based on throughput obtained and price charged, was proposed in [19]. However, the work does not take into account the existence of heterogeneous networks or the characteristics of traffic, and does not specify an operational method to split the traffic. Our work, on the other hand, accounts for all these aspects.

An analytical framework for allocation of services (e.g. voice and data) to multiple radio access technologies in order to maximize the combined multi-service capacity is presented in [11], and in [16] the authors examine algorithms for access selection by drawing a parallel with bin packing problems with the bins representing the access networks into which user services have to be packed. It is assumed in [11], [16] that the radio access networks are operated in a coordinated fashion. The suggested service allocation strategies represent a network-centric approach for resource allocation and do not touch upon technology specific implementation issues for executing the service allocation measures. Furthermore, the allocation of services to networks is static and is not dynamically varied according to varying network characteristics. Our work does not require any changes in or coordination between heterogeneous network access technologies that a device has access to, and suggests measures that can be employed by the device to dynamically assign traffic flows to the access networks.

Flow scheduling for collaborative Internet access in residential areas via multihomed client devices is discussed in [22]. The scheduling framework proposed in the work only accounts for TCP flows and uses metrics useful for web traffic including RTT and throughput for making scheduling decisions. Our work on the other hand is generic and uses the stochastic characterization of networks and maximization of rewards offered by access networks to the flows with given utility functions for making flow scheduling decisions. We demonstrate the performance benefits of our flow assignment framework by employing elastic video flows with concave utilities.

In a recent work [14] on optimal user-network association, an MDP based framework has been used for routing arriving mobile user's connection to one of the available WLAN or UMTS networks. The entire traffic generated by a user is treated as a single entity and by virtue of the routing decision the user device connects to one network at a time. However in a real world scenario, a multi-homed device can have connectivity to multiple networks and it is a challenge to efficiently route the user traffic onto these networks. Our work addresses these problems and uses traces from real networks to demonstrate the benefit of MDP based flow assignment.

In some of our own recent works we have looked at rate control [5], video streaming [26] and middleware architectures [20] for heterogenous network access. The present work focuses on flow assignment and interface selection problems in multi-homed networks.

3. System Model and Analytical Framework

Fig. 1 depicts the operational scenario for routing of flows originating from applications running on a device via access networks that the device has access to. The system components of the device include a *middleware* functionality that runs a lightweight tool to estimate the ABR and delay via different access networks to the destination hosts in the Internet. ABR and delay constitute key network characteristics that ascertain the performance of a wide variety of applications. While low latency is critical for interactive and short-lived TCP flows, the performance of long-lived flows like bulk-transfer TCP flows depends the rate available in the access network. The performance of video streaming sessions is governed by both ABR and delay as we will discuss later in this section.

Applications running on the device consult the middleware for routing of flows. The list of preferred destinations hosts can be maintained at the device based on user usage history, user preferences, etc., as for instance described in [12].

We denote the set of access networks available to the device by $I = \{1, 2, \ldots, N\}$. The system state, designated as $s \in S$, represents the delay and ABR characteristics of all the networks. The objective then is to decide an optimal policy $\mu^*(s)$ that, as function of system state, suggests action $u \in I$ that signifies a network to be selected for the assignment of a flow. The system evolution is modeled as a discrete time process and the optimal policy is evaluated at the beginning of fixed time slots for the assignment of a given flow in the system to a suitable network. At a given decision epoch, the system decides the interface that maximizes a discounted reward for a flow. The state of the system is as observed through active measurements on all access interfaces. The decision is consulted at a time epoch by a flow in the system and that flow is accordingly assigned to an optimal interface.

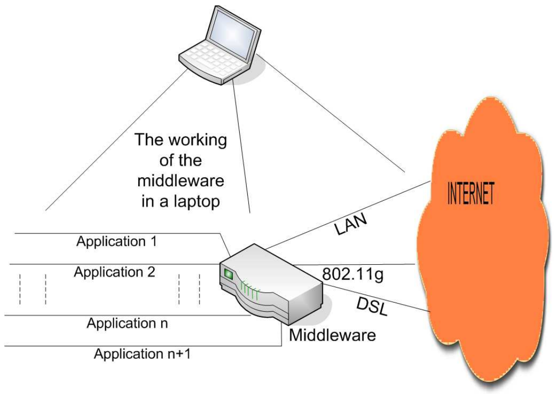

Figure 1. Middleware functionality in a device

The interface assignment is performed for the flows existing in the system in a round robin fashion in successive time slots. Our model avoids excessive switching overhead by analyzing the system as observable by a given flow, and only reassigning that flow at a decision epoch. We note that flow reassignment comes at the cost of system and network overhead. For a multi-homed device additional processing and memory capability needed for switching may not be available especially for the case of light weight mobile devices. Moreover, additional processing per switched flow is also required on the part of the network. A proxy server typically needs to ensure that the data from a flow are routed to the correct destination irrespective of the networks they traverse. As these proxies would potentially handle millions of flow in a real world scenario, too frequent switching of a flow per device comes at an expense of having to deploy additional servers.

Each network $i \in I$ is characterized by delay and ABR values $d_i \in [d_i^{min}, d_i^{max}]$ and $r_i \in [r_i^{min}, r_i^{max}]$ measured at a given time using an online measurement tool. We map the range of delay and ABR values of each network to a set of quantized states. Let $S_d^i := \{s_{d1}^i, \ldots, s_{dK}^i\}$ and $S_r^i := \{s_{r1}^i, \ldots, s_{rL}^i\}$ represent states based on the quantized delay and ABR of the network i. Then each network is associated with a single superset of states $S^i := S_d^i \times S_r^i$ and the whole system with $S := \prod_{i \in I} S^i$ obtained through cross-product operations on the sets.

The delay and ABR of each network exhibits variations due to a multitude of underlying factors ranging from fading and shadowing in wireless channels to cross-traffic and congestion in wired ones. While the wired access networks may be characterized using simpler techniques, the fast variations in wireless access network characteristics require robust stochastic models. It is shown in the literature that Markov models can well characterize network characteristic variation behavior[13, 18]. In addition, there exists well-established computational and theoretical methods to optimize Markovian processes [6]. Hence, we define a finite-state Markov chain on the state space S to model the system at hand. We denote by $p_{i,j}$ the transition probability

$$p_{i,j} := P(s(n+1) = j \mid s(n) = i), \tag{1}$$

where $i, j \in S$ and $s(n)$ represents the current state of all available networks at time step n. Consequently, the state transition matrix is defined by $M := [(p_{i,j})]$ and the system equation is

$$\tilde{p}(n+1) = \tilde{p}(n)M, \tag{2}$$

where $\tilde{p}$ denotes the probability vector over the state space S corresponding to all possible states of multiple heterogeneous networks.

There exist a variety of methods for computing the transition probabilities between network states. We make the implicit assumption of ergodicity and stationarity over a certain time interval over which M is time-invariant. Given sufficiently many state transition pairs obtained from the evolution of network characteristics over time it is possible to compute M using standard methods [10]. Let us use the first state of each pair $C^{in} = \tilde{p}(n)$ where n is even, as initial conditions for the underlying dynamical system and denote $C^{out} = \tilde{p}(n)$ where n is odd as the image of these points after one iterate of the dynamical system. After identifying the sets of input and output samples C^{in} and C^{out} the transition probability from state i to j is estimated as

$$\hat{p}_{ij} = \frac{\sum_{[k:C_k^{out} \in j]}}{\sum_{[l:C_l^{in} \in i]}}, \tag{3}$$

where $\sum_{[k:C_k^{out} \in j]}$ denotes the number of points k such that $C_k^{out} \in j$.

As the number of state-transition pairs increases (i.e., as $n \to \infty$) the invariant measure of the Markovian operator M approximates the time-averaged distribution of the states better.

The control action u corresponding to the choice of a single network modifies the dynamical system and leads to control Markov chains $M(u)$. The transition probability $p_{ij}(u_k)$ of the controlled Markov chain denotes the probability of the next state being in j conditioned on the current state being in i and control being u_k. $M(u)$ for a given u has a dimension $S \times S$. Note that u assumes a value in I corresponding to the choice of a network.

Now that we have a Markov model on a finite state space S with finitely many control actions u, we pose the control problem at hand as an MDP. Towards this end let us define the real-valued reward function $R(s, u)$ over the set of states S and as a function of the control action u. The reward function quantifies the preference for a system state and the choice of a network by a flow. Depending of the type of application, the reward function can have different characteristics. For instance, the flow-assignment framework can be applied to applications based on bulk-transfer TCP or multimedia traffic. Multimedia traffic can especially benefit from efficient utilization of resources available on multi-homed devices since the user experienced quality tends to be very sensitive to the bit rate and delay available to the flows. In this work, we thus formulate the reward function to represent the reward offered by a network to flows which exhibit the characteristics of video traffic:

$$R(s,u) = f(r^u)\ u_s(r^u - r_{min})\ u_s(T - d^u), \tag{4}$$

where $f(r^u)$ is a concave utility function, and $u_s(r^u)$ and $u_s(d^u)$ are unit step functions. The flows represented in Equation (4) are characterized by a minimum usable bit rate r_{min}. For rates greater than r_{min}, the utility function for the flows is concave with respect to the allocated bit rate. Furthermore, the packets belonging to the flows are associated with a maximum latency (or the playout deadline) T that they can tolerate. For $f(r^u)$ in Equation (4), we employ video encoder rate distortion models from [21] and adopt the following form:

$$f(r^u) = 10log_{10}(255^2/D(r^u)), \tag{5}$$

which represents the Peak Signal to Noise Ratio (PSNR) of an encoded video stream with encoder distortion $D(r^i)$ given by

$$D(r^u) = \frac{\theta}{r^u - r_0} + D_0. \tag{6}$$

The parameters θ, D_0, and r_0 can be estimated [21] from empirical rate-distortion curves via regression techniques. $\hat{R}^u(r^u, d^u)$ is plotted in Fig. 2 for a representative scenario.

Now the reward per stage in the MDP framework, $g : S \times I \rightarrow \Re$, under a stationary policy $\mu(s) :\ S \rightarrow I$ and at stage n is given by $g(s(n), \mu(s))$, with $\Re$ representing the set of real numbers. Notice that for a system state s and a given stationary policy $\mu(s)$, $g(s, \mu(s))$ is equal to the reward $R(s, u)$ with $u = \mu(s)$.

Although the stationarity window for the policies may be limited we assume an infinite horizon formulation of the problem as a simplification. Then the total reward J is given as :

$$J_\mu(s) := \lim_{N\to\infty} \sum_{n=0}^{N} \alpha^n g(s(n), \mu(s)), \tag{7}$$

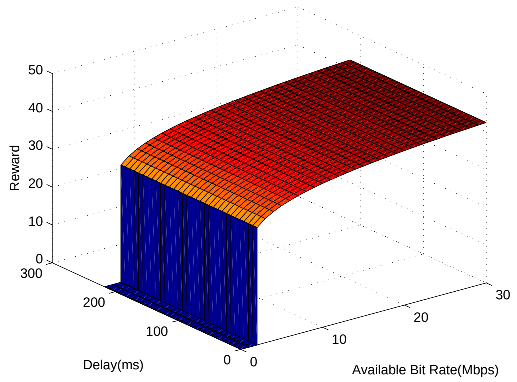

Figure 2. Reward (with units of PSNR (dB)) offered by a network to an incoming flow for $\theta = 97.8$, $r_0 = 0.075$ (in Mbps), $D_0 = 0.49$, $T = 150ms$, and $R_{min} = 2Mbps$

where the positive scalar $0 < \alpha < 1$ denotes the discount factor over future stages (decisions). The maximum total reward is defined by

$$J^*(s) := \max_{\mu \in \Pi} J_\mu(s), \quad s \in S, \tag{8}$$

where Π is the set of all possible policies. We say that the policy $\mu(s)$ is optimal if $J^*_\mu(s) = J_\mu(s)$ for all states s. It is a well known fact that under certain assumptions ([6], Chapter 1) there exists an optimal deterministic stationary policy $\mu^*(s)$ that solves the *Bellman's equation*, i.e.,

$$J_{\mu^*}(s) = g(s, \mu^*(s)) + \alpha J_{\mu^*}(s\, M(\mu^*(s))), \quad \forall s \in S. \tag{9}$$

Furthermore, $J^* = J^*_\mu(s)$ is the unique solution of Bellman's equation.

There are multiple alternative algorithms to solve the infinite horizon discounted reward problem described above. In this paper, we choose without loss of generality the well-known *value iteration* algorithm. The right hand side of the Bellman's equation in (9) actually corresponds to a single iteration or recursion of the venerable dynamic programming algorithm (DPA)

$$J_{n+1}(s) = \max_u g(s(n), u) + \alpha J_n(s(n) M(u)), \tag{10}$$

at the time step n and state $s \in S$. The value iteration algorithm is based on the fact that the DPA converges to the optimal reward

$$J^* = \lim_{n \to \infty} J_{n+1}(s), \tag{11}$$

and can be obtained simply by turning the Bellman optimality equation into an update rule. For a detailed discussion on the topic we refer the reader to [6].

The value iteration algorithm as described above is executed and updated periodically by the middleware on a device to evaluate the control policy $\mu^*(s)$ for the assignment of a flow to an access network. The middleware maintains information about the current flow assignment and every time slot gives a flow on the device a token which is circulated in a round robin fashion amongst flows in successive time slots. Every decision epoch the flow which gets the token seeks middleware consultation and is assigned to an access network according to the control policy. The access network states are ascertained from the online ABR and RTT measurements made by the middleware. We note that the measured ABR assesses the unused bit rate on an interface. The rate allocated to a flow before the current assignment is added to the ABR measured on the interface on which it existed before. This determines the state of this interface from the perspective of the flow with the token seeking optimal assignment. Thus the effective ABR on this interface is taken as the sum of ABR noted via active measurements and the rate assigned to the flow on that interface in the previous time epoch. The effective ABR on other interfaces is taken as the ABR noted via active measurements. On interface assignment, the flow is provided a rate which is half of the ABR on the interface to which it is assigned.

Each MDP invocation is preceeded by a *training* wherein the system state transitions associated with the control action are monitored and $M(u)$ is thus evaluated.

During a time slot, the characteristics of an access networks may change. The flows continue to have the rates from the beginning of the slot. When flow assignment decision is made at the next epoch, new characteristics of the networks obtained via online measurements are used for MDP evaluation and flow assignment.

We note that the complete characterization of an optimal MDP based policy that maximizes the discounted reward of a given flow would require information on several parameters including the network to which the flow is assigned, the rate assigned to the flow, the identity of the flow with the token, etc. Since the assignment of this flow is impacted by the remaining flows in the system, the corresponding information and the rewards for these remaining flows would also need to be incorporated in the model. While such an approach would yield the optimal solution, it would be ridden by the well known state space explosion problem.

To keep the dimensionality tractable and the execution of the optimization feasible, we in our work model the framework from the perspective of a flow that seeks assignment from the multi-homed system having connectivity multiple access networks. Every time epoch such a flow is chosen via the circulating token policy. This flow is treated as one that has appeared in the multi-homed system and needs to be assigned to a suitable access network. The remaining flows in the system are considered as background traffic. Via online measurements, we observe the characteristics of the system as would be seen by the flow with the token. We discount the details pertaining to flows other than the one with the token. We will further discuss the details of MDP execution in Section 5.

We compare the above MDP based flow assignment policy described above with static and dynamic flow assignment schemes:

- Static Flow Assignment: The flows are admitted with a rate equal to the r_{min} for their class. Thereafter the flows exercise a TCP-style Additive Increase Multiplicative Decrease (AIMD) policy for rate control. An interface-specific token is circulated and a flow on each interface having this token increases its rate by Δ_r unless network congestion is perceived by a flow in which case it drops its rate by $(r - r_{min})/2$. We note that the dropping of rate above the minimum by two is motivated by the congestion control dynamics of TCP which halves its window upon indication of congestion. TCP like congestion control continues to be a popular mechanism for both HTTP and non-HTTP based media streaming over the Internet. We employ two static flow assignment policies - one where the flows are assigned to networks in a greedy fashion and the other in proportion to the average ABRs of the networks. Under the former policy, a flow is assigned to a network that offers the maximum instantaneous reward (as given by Equation (4)) to a flow upon its admission. We term this policy as greedy-static. Under the other static assignment scheme, flows are allocated to different networks in proportion to the average ABR on the networks. In other words the flows are probabilistically assigned to the networks with assignment probability proportional to the average ABR on the network. We call this assignment scheme Rate Proportional static (RP-static).

- Dynamic Flow assignment: At each time epoch a flow with a token circulated by the middleware is dynamically assigned to an interface which offers the maximum ABR to the flow. The flows uses AIMD rate control as explained for the static case. We call this flow assignment greedy-dynamic.

4. Network measurements

In this section we present results from network measurements conducted in a real world setting. Employing the modeling framework of the previous section, we will use the measurement traces to simulate and evaluate the flow assignment framework in the subsequent sections.

We conduct measurement tests in a corporate work environment where the users have access to networks like Ethernet, IEEE 802.11g and IEEE 802.11b WLANs, GPRS, and DSL. We monitor the ABR and RTT on different networks between 2 PM and 4 PM on a work day. The tests are conducted between hosts in Deutsche Telekom Laboratories (T-Labs) in Berlin to three destinations - Stanford University, Technical University of Munich (TU Munich), and the Technical University of Berlin (TU Berlin) - respectively representing long, mid, and close distance destinations. We surveyed several publicly available tools including Pathrate, Nettest, CapProbe and choose Abing [17] for measurement of ABR and round trip time (RTT). Abing has a fast convergence of the order of 1-2 seconds, is lightweight, and has the ability to run accurately on paths with high packet loss rates, and is hence reported [17] to be suitable for wireless networks. It is based on packet pair dispersion technique and reports the ABR for bidirectional links between two hosts in the Internet which run Abing client and server. The underlying principle for measurement is to send packet probes to destination and measure the inter-packet delay as they arrive at a destination. Abing uses packet pairs having fixed size and sends several such closely spaced probes. The round trip time is reported by the tool as the train of packets in sent in each direction to measure the bi-directional ABR. We run Abing server at machines at Stanford, TU Munich and TU Berlin connected to the Internet via high speed LANs, and the client machines at T-Labs in Berlin connected via different access networks. The ABR and RTT values are then noted every second for the links from T-Labs and to different destinations. For the purpose of this work we consider the data collected on 100 Mbps Ethernet, IEEE 802.11g, and IEEE 802.11b networks. The 802.11g and 802.11b networks were accessed by laptops with Intel PRO/Wireless 2200 b/g cards through T-Sinus 154 and linksys WRT-54GL wireless access points (APs) respectively.

The test environment represented a well provisioned wireless LAN setting with 5 APs in a large office room. The measured networks had interference from other APs in the room and also APs from the higher and lower floors in the building. Tables I, II and III show the average ABR and RTT and their standard deviations to different destinations and for different networks for the 2 hour traces. Ethernet can be seen to

Table I. Available bit rate and RTT from T-Labs to Stanford University

Network	Statistics	ABR(Mbps)	RTT(ms)
Ethernet	Avg.	31.5	190.1
	Std. Dev.	1.7	0.03
802.11g	Avg.	15.1	193.0
	Std. Dev.	3.6	3.2
802.11b	Avg.	4.2	195.7
	Std. Dev.	0.3	0.3

Table II. Available bit rate and RTT from T-Labs to TU Munich

Network	Statistics	ABR(Mbps)	RTT(ms)
Ethernet	Avg.	90.8	14.4
	Std. Dev.	6.0	0.1
802.11g	Avg.	15.0	16.9
	Std. Dev.	3.8	4.5
802.11b	Avg.	4.4	19.8
	Std. Dev.	0.4	1.0

have different ABRs to different destinations which can be attributed to different cross-traffic and intermediate bottleneck link capacities to these destinations. However, the average bit rates to different destinations are not much different for 802.11g and 802.11b indicating the possibility that ABR is constrained by the bottleneck wireless hop. RTTs to a destination are lowest for Ethernet and highest for 802.11b.

Figs. 3, 4, and 5 show representative histograms of the ABRs for the destination Stanford. The statistics can be seen to have diversity in ABRs across the three networks (the average ABR on Ethernet can be seen from Table I to be twice as much as on 802.11g which is roughly four times as much for 802.11b). All the networks display noticeable variation in ABRs. For instance the ABR on 802.11g can be as high as 24 Mbps and can drop down to as low as 6 Mbps.

The different ABRs on the networks reflect the difference in the ability of these networks in accommodating traffic flow volumes. Flows may be assigned to the networks according to their ABRs. However, as the characteristics of a given network fluctuate (for instance when there

Table III. Available bit rate and RTT from T-Labs to TU Berlin

Network	Statistics	ABR(Mbps)	RTT(ms)
Ethernet	Avg.	71.8	5.2
	Std. Dev.	13.0	0.04
802.11g	Avg.	14.3	7.8
	Std. Dev.	3.6	0.4
802.11b	Avg.	4.5	10.7
	Std. Dev.	0.5	0.6

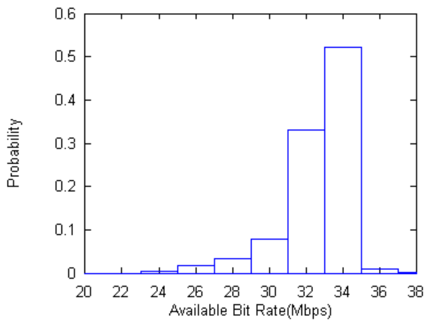

Figure 3. Available bit rate on Ethernet from T-Labs to Stanford

are abrupt drops in ABR), the supported applications may suffer from performance degradation. Then, if some of the flows under adverse network conditions can be directed to another network, the performance of the applications and utilization of the networks can be improved. We will investigate this further in the Section 5.

We noticed that the scale of variation of ABR and delay was much greater for the wireless networks than for Ethernet, which justifies the use of MDP based stochastic modeling over a simpler approach when the access environment includes wired and wireless networks. For instance the average interval of variation of ABR by 10% was 10 times higher for 802.11b and 3 times higher for 802.11g than the ABR variation over Ethernet for T-Labs to Stanford case.

5. Performance Evaluation

We simulate the flow assignment framework using ns-2. The sample network topology created for the purpose is shown in Fig. 6. The node-S represents the sending device which sends flows to destination node-D

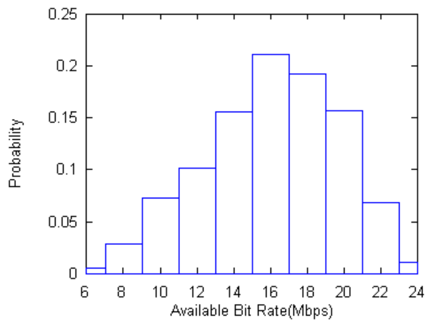

Figure 4. Available bit rate on 802.11g from T-Labs to Stanford

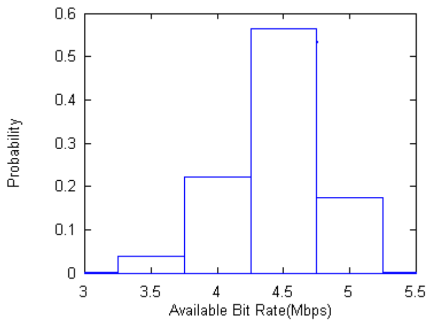

Figure 5. Available bit rate on 802.11b from T-Labs to Stanford

via the networks N1, N2 and N3 using its middleware. We describe the functionality of the components and the tools employed below.

1. **Simulation of Access Networks** : Each network (e.g. N1, N2, and N3 in Fig. 6) is simulated as a link with varying available bandwidth and delay characteristics. These characteristics are obtained from the practical measurements performed in real networks settings - e.g. the ones described in Section 4.

2. **Flow Assignment** : An instance of hash classifier [2] is attached to a node performing flow routing and is used to simulate a *broker* whose function is to direct various flows to different networks based on the policy calculated by the middleware. We implement part of the middleware functionality by interfacing python functions with ns-2. The middleware for a device measures the ABR and delay on the different networks, and performs the flow assignment using MDP. The flows at a node are identified via *flow id*s. We ensure that the broker agent attached to the node has information about every flow generated from the source and coming to the source from the Internet.

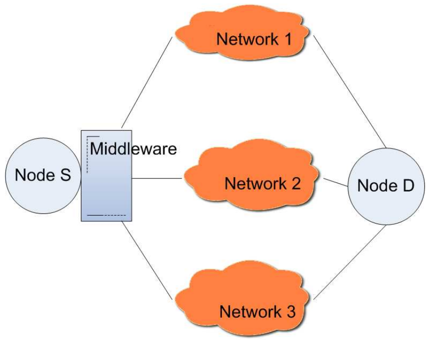

Figure 6. Network topology in ns-2

3. **Network delay measurement** : We employ CapProbe [15] implementation for ns-2 to calculate RTT of networks. For this purpose, we attach a ping agent for every network to be monitored (e.g. N1, N2, N3 in Fig. 6) to the node (e.g. Node-S) performing flow assignment and associate every ping agent with a flow id to be used by the hash classifier for routing the ping traffic.

4. **ABR Measurement** : The network bandwidth utilized at a given time is measured during the simulation via queue monitors [2] attached to the links corresponding to the networks. The number of bytes transferred via the link during a 0.1 second interval is used to calculate the used bandwidth. ABR during the simulation is periodically evaluated by subtracting the network bandwidth being used from the present value of ABR used to characterize the network. In real world scenarios, tools like Abing can be used to measure ABR.

For the demonstration of evaluation results, the three networks shown in the ns-2 topology of Figure 6 are taken as Ethernet, 802.11g, and 802.11b with ABR (r) and delay (d) characteristics of Section 4. The delay d is approximated as half of RTT values measured for different networks. Simulations are run over the 2 hour data traces for different destinations. For the 802.11b and 802.11g wireless networks we introduce a 1% random packet loss in the simulations.

We employ high bit rate flows with the characteristics of Section 3 with $r_{min} = 2$ Mbps and $T = 150$ ms. At the beginning of a simulation, a total of 14 flows arrive during the arrival phase with a rate of 2 Mbps each, following a Poisson process with an interarrival time of 10s. Subsequently the rates of the flows evolve as per the rate control associated with the employed flow assignment policy. The middleware monitors ABR and RTT to the destination hosts via each network periodically.

For the greedy-static policy, a flow upon arrival is allocated to a network that offers the maximum instantaneous reward given by Equation (4). For RP-static flow assignment, flows are allocated to networks in proportion to the average ABR reported in Tables I, II, and III. For both these static policies, the bit rate of each flow is varied according to a token-based round-robin scheme where the token is circulated every 2 seconds. The round robin scheme operates separately and independently on each network. The flow with a token on a given network increases its flow rate by $\Delta_r = 1$ Mbps and passes the token to the next flow on the network. Whenever a delay rise beyond the preset threshold, $d_t = 150$ ms, is observed (by CapProbe measurement tool) for any of the networks, a delay alarm for that access network is triggered, and the bit rate for the flow is reduced by $(r - r_{min})/2$.

For the greedy-dynamic policy, a flow upon its arrival is allocated to an interface that offers the maximum ABR. A round robin scheme is then followed to circulate a global-token amongst the existing flows on all interfaces. At a given time epoch, the flow with this token is dynamically assigned to an interface that offers the maximum ABR to the flow. Flows on an interface also carry an interface-specific token for AIMD rate control as described above for the static policies.

For the MDP based flow assignment, when the flows initially arrive, they are greedily allocated to the networks based on the maximum reward. Thereafter, the reassignment and rate allocation of the flows is done via a round robin token scheme with a token circulation interval of 2 seconds. Every time during a simulation a flow gets a token, the optimal MDP policy in the middleware is consulted by the broker, the control action suggesting a suitable interface for the flow is executed, and rate allocation for the flow is performed.

The decision epoch for executing the MDP policy is taken as 2s. The ABR and delay values on the interfaces are quantized. The quantization level for ABR on the interfaces is taken to be 2 Mbps. The Ethernet Interface has 50 ABR quantization levels, IEEE802.11g has 15, and 802.11b has 3 levels. The delay on each interface is quantized to 4 levels with a quantization interval of 40 ms. Hence the state space on each of these interfaces has a cardinality of 200, 60 and 12 respectively. For evaluating the MDP policy every time epoch, the middleware needs information about the system state transitions via-a-vis the control actions. For the purpose we maintain a sliding window of 300 transitions and action combinations amounting to a time window of 600s.

The value iteration for solving the MDP framework described in Section 3 is outlined in Algorithm 1. The algorithm evaluates the optimal policy $\mu(s)$ for all states s in the state space S. Lines 3-12 represent the DPA recursion on Equation 10. However, with the above discussed

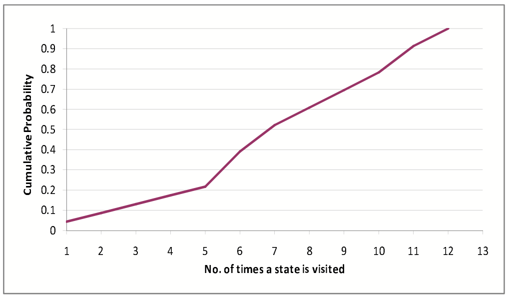

Figure 7. Visited state space on Ethernet (T-Labs to TU Berlin)

Algorithm 1 Value Iteration with output $\mu(s)$ $\forall$ s $\in$ S

Initialize: $V(s) \leftarrow 0 \; \forall \; s \in S$
Initialize: $\delta \leftarrow 0$
repeat
  $v \leftarrow V(s)$
  for all $s \in S$ **do**
    for all $u \in I$ **do**
      $Q(s,u) \leftarrow R(s,u) + \alpha \sum_{s' \in S} M(s,u,s')V(s')$
    end for
    $V(s) \leftarrow \max_u Q(s,u)$
  end for
  $\delta \leftarrow \max\{\delta, |v - V(s)|\}$
until $\delta < \theta$
for $s \in S$ **do**
  $\mu(s) \leftarrow \arg\max_u Q(s,u)$
end for

quantization levels of the interfaces and the execution methodology, the system would have a total of 144,000 states. This explosion of state space makes the training needed for $S \times S \times I$ state transition and action combinations difficult. Moreover, in a real world scenario there are often limitations on memory, processing and battery power of user devices.

We reiterate the rationale as discussed in Section 3 where we focus on the MDP execution from the perspective of a given flow seeking middleware consultation. The flow with the token is treated as the one that has appeared in the multi-homed system and needs to be assigned to a suitable access network by virtue of the control action. In our model, we discount the characteristics of other flows which are taken to constitute background traffic. The details pertaining to identity of flow with token, the interface to which the flow is assigned, etc. are not directly modelled in the MDP framework.

We note that the decision to be made in a given time epoch is regarding the interface that can optimally host a flow over a period that includes future time epochs since the flow stays on an interface until it receives the token again. We thus make an approximation and discount the change in states of other interfaces while evaluating the DPA recursion and target choosing the interface that offers the best discounted reward to the flow with the token. As we will see, this greatly reduces the dimensionality of the state space. In a real-world setting, the access networks are usually independently provisioned (e.g. 802.11b and 3G) and hence the traffic on one does not impact the other.

In a model that accounts for complete information on the state of the flow being assigned and the other flows in the system (as discussed in Section 3), the above assumptions would not hold accurate. However in the approach adopted in this work, we show below that these assumptions lead to substantial reduction in complexity and significant performance enhancement over conventional interface assignment policies in multi-homed environments.

We thus disentangle the state space to reduce its dimensionality. In Algorithm 2 we highlight the value iteration performed during the simulations. The control action u designates assignment to an interface in I. The complete state space S is given by $\bigcup_{i \in I} S^i$. The value of the discount factor α is taken as 0.7.

Algorithm 2 Value Iteration with output $\mu(s)$ $\forall$ s $\in$ S

```
Initialize: V^u(s^u) ← 0 ∀ u ∈ I, s^u ∈ S^u
Initialize: δ ← 0
repeat
  for all u ∈ I and s^u ∈ S^u do
    v^u ← V^u(s^u)
    V^u(s^u) ← R^u(s^u) + α Σ_{k∈S^u} M^u(s^u, k) V^u(k)
    δ ← max{δ, |v^u − V^u(s^u)|}
  end for
until δ < θ
for s ∈ S do
  μ(s) ← arg max_u V^u(s^u) /* Note that s^u ⊂ s */
end for
```

As a consequence the number of system states are reduced to sum of the states of individual interfaces (272). The number of state transition and action combinations then amount to $\sum_i |S^i \times S^i|$. The number of these combinations is still loo large to be trained. However, only a small fraction of these combinations are relevant for training purposes as a predominant number of state existence and state transitions are improbable. For instance existence of a state with high ABR and high delay to a local destination is unlikely as when the bit rate available on an interface is high its delay would be low. Again, the transition from a state with moderate ABR to low ABR on an interface upon flow assignment to that interface is not possible as the flow assignment would only reduce the ABR on the interface. To quantify this better and support our claim we report statistics from our simulations. For T-Labs to TU Berlin case, the number of states visited during a simulation run on the Ethernet interface is only 23 in number and on an average a state is visited 7.39 times during a state transition and action training window of 600s. Fig 7 plots the distribution of the number of times a state is visited on an average during a window of 600s in a simulation run. As can be seen 80% of the states are trained 4 or more times in a window. Similar trends prevail on 802.11g and 802.11b interfaces and for Munich and Stanford destination network scenarios. When the system is found to be in a state not visited before, we fall back to the greedy approach. The assignment of the flow is done to an interface that offers the maximum reward to the flow.

Before the MDP based flow assignment can be invoked for the first time by the middleware, MDP needs a training period to evaluate $M(u)$. We hence train the MDP in the middleware for initial time window with u based on a reward-maximizing greedy approach for network selection. The rate allocation during this phase is kept the same as that discussed in Section 3 under MDP based flow assignment, i.e. if as a consequence of a control action a flow is reassigned to a different network than it was previously on, then it is provided a rate which is half of ABR on the new network, otherwise the rate of the flow is increased by half of ABR of the present network.

With the simulation methodology described above, we compare the performance of the static and dynamic flow assignment policies with MDP based approach. Fig. 8, 9, and 10 show the average flow rates during the simulation runs for all the flow assignment schemes for T-Labs to TU Berlin, TU Munich and Stanford cases respectively. As can be seen for the plots, MDP based flow assignment leads to the highest average flow rates amongst all policies.

Figures 11, 12, 13, and 14, show the time windows during simulation run with percentage utilization of the access networks for different flow

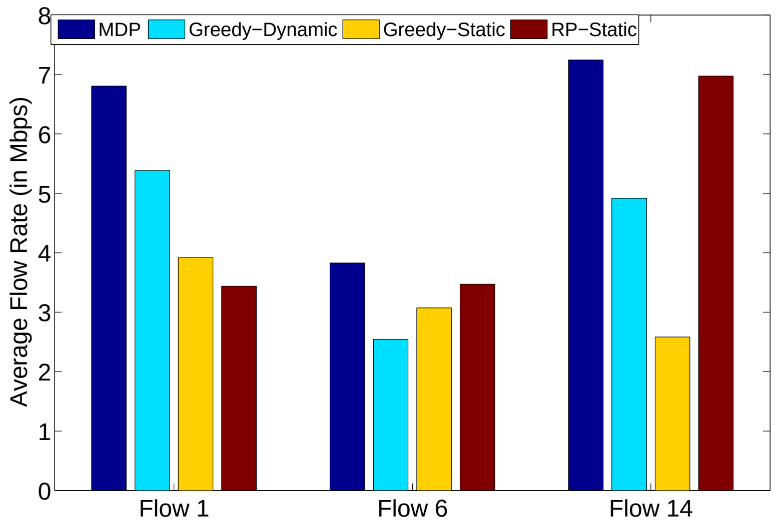

Figure 8. Average flow rate (T-Labs to TU Berlin)

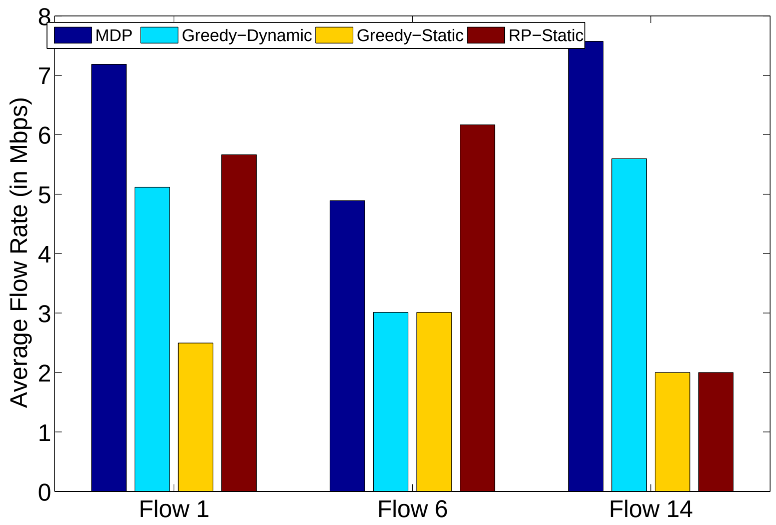

Figure 9. Average flow rate (T-Labs to TU Munich)

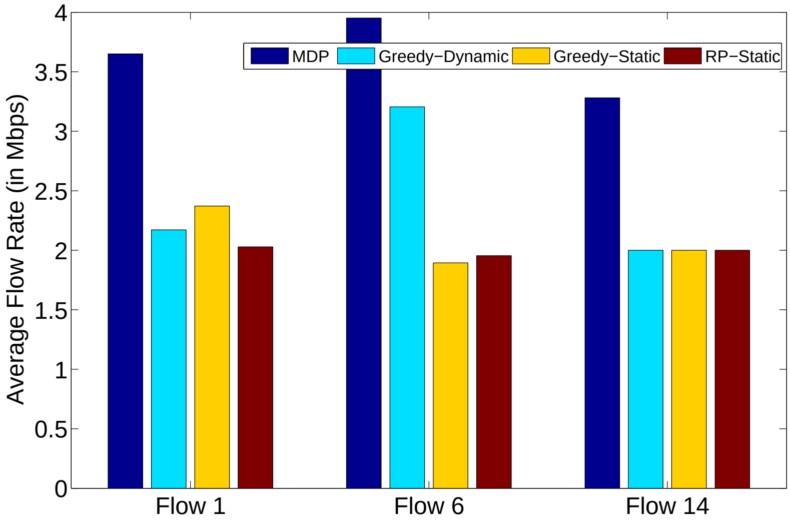

Figure 10. Average flow rate (T-Labs to Stanford)

assignment schemes for TU Berlin case. The greedy-static policy leads to assignment of all the flows to the Ethernet interface and hence all other access networks remain unutilized. The average utilization of the access interfaces is plotted in Fig. 15. The TU Munich and Stanford scenarios exhibit similar trends.

Tables IV, V, VI show the average of packet loss rates and flow rates. As can be seen, MDP based flow assignment policy leads to highest average flow rates and significantly low packet loss rates. For instance, for the Stanford case MDP lowers the packets loss rate from around 43% for greedy-static, 46% for RP-static, and 53% for greedy-dynamic to a significantly lower 8%.

Table IV. Packet loss rate (PLR) and flow rate statistics for Stanford University

	Greedy Static	RP Static	Greedy Dynamic	MDP
Avg. PLR(%)	42.16	46.26	52.74	8.25
Avg. ABR(Mbps)	2.08	1.99	2.45	3.6

The MDP based flow assignment leads to significantly lower delays on all three networks. The *cdf*s of the delays for all flow assignment schemes for TU Berlin case are plotted in Figs 16, 17, and 18. The delays for Stanford and TU Munich cases show the same trends.

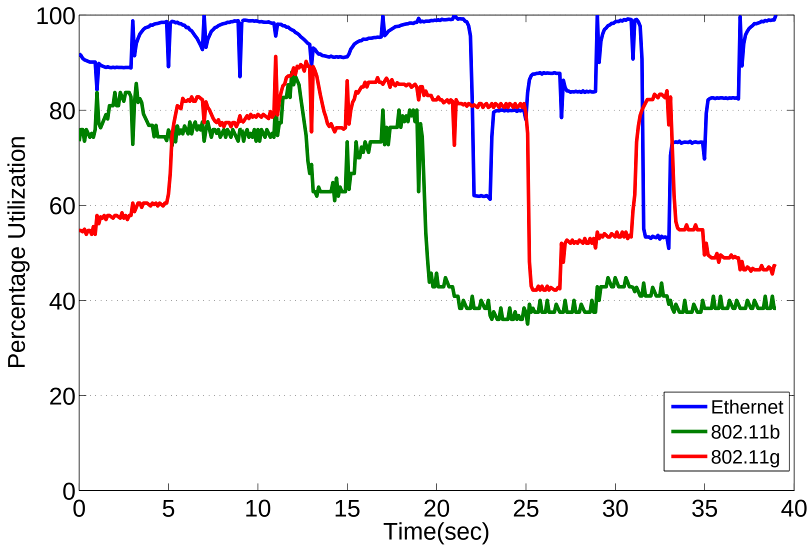

Figure 11. Percentage utilization for MDP (T-Labs to TU Berlin)

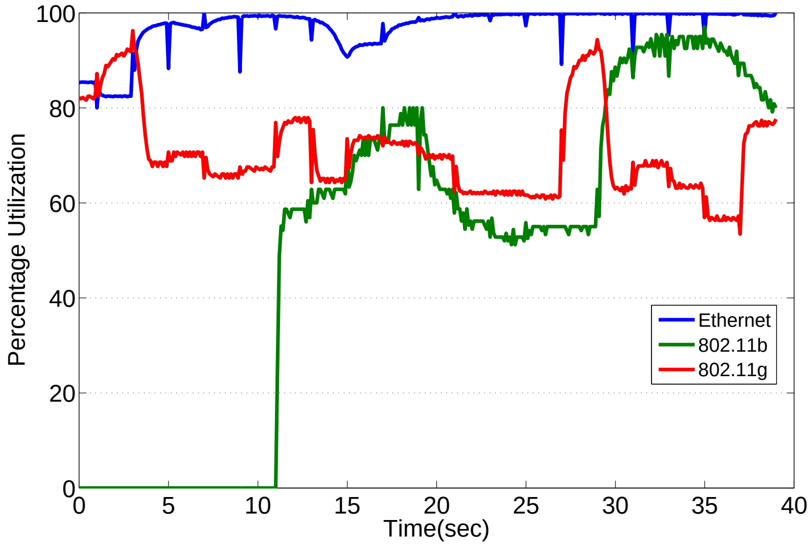

Figure 12. Percentage utilization for greedy-dynamic (T-Labs to TU Berlin)

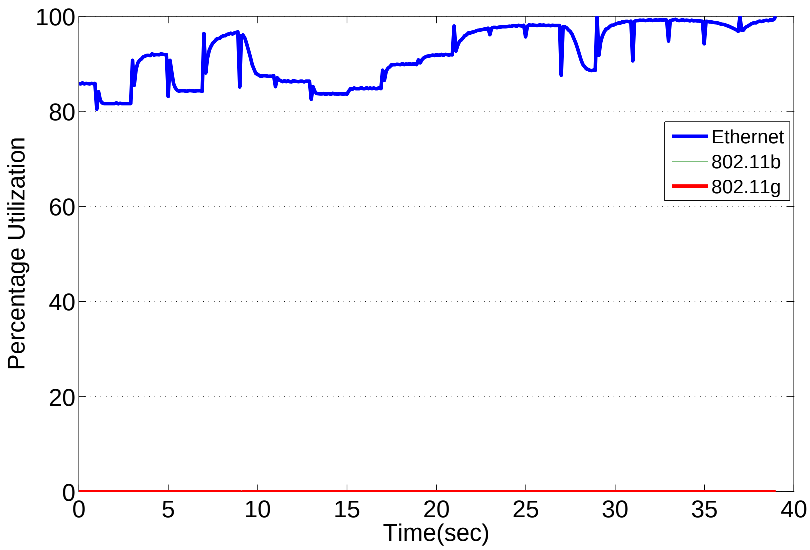

Figure 13. Percentage utilization for greedy-static (T-Labs to TU Berlin). All 14 flows are allocated to Ethernet and hence utilization of 802.11g and 802.11b networks remains 0.

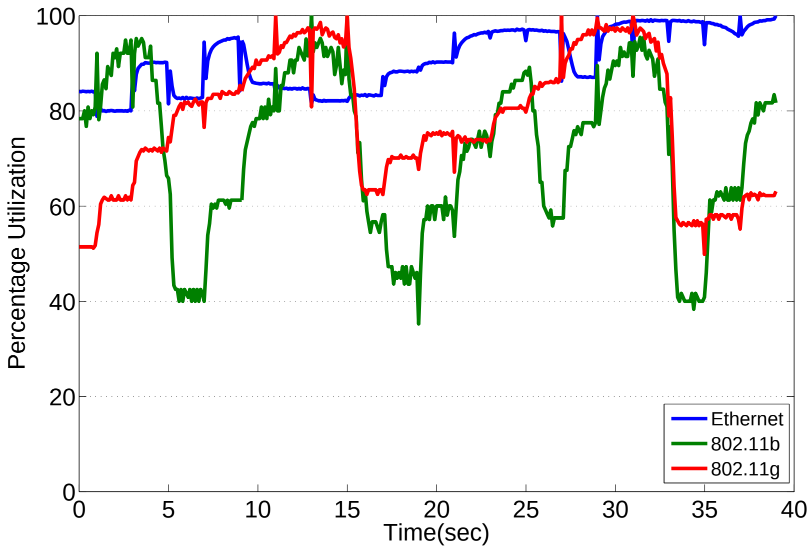

Figure 14. Percentage utilization for RP-static (T-Labs to TU Berlin)

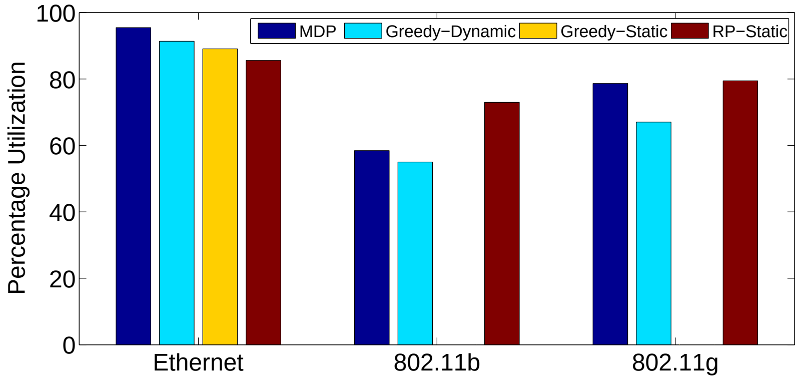

Figure 15. Average percentage utilization (T-Labs to TU Berlin). The utilization of 802.11g and 802.11b networks for greedy-static policy remains 0 as all the flows are assigned to the Ethernet interface by the policy.

Table V. Packet loss rate (PLR) and flow rate statistics for TU Berlin

	Greedy Static	RP Static	Greedy Dynamic	MDP
Avg. PLR(%)	35.35	33.93	52.44	19.56
Avg. ABR(Mbps)	3.19	4.6	4.28	5.95

The average PSNR of the flows for different flow assignment schemes during a time window in the simulations is shown in Fig. 19. As can be seen from the plot, MDP results in best PSNR performance for the flows.

The impact of variation of playout deadline (T) on the packet loss rate performance is shown in Table VII. Increase in the deadline leads to significant reduction in packet loss rate for MDP based flow assignment, but does not have as noticeable an impact for the greedy-dynamic, greedy-static and RP-static policies.

6. Conclusion

Multiple network utilization via a flow allocation policy which stochastically characterizes the network characteristics and dynamically assigns flows to the networks results in significantly enhanced performance over static and dynamic policies that assign flows based on greedy approaches or heuristics like average available bit rate on the networks. Even in conjunction with suitable rate control schemes, ordinary static or dynamic flow allocation policies suffer from degraded

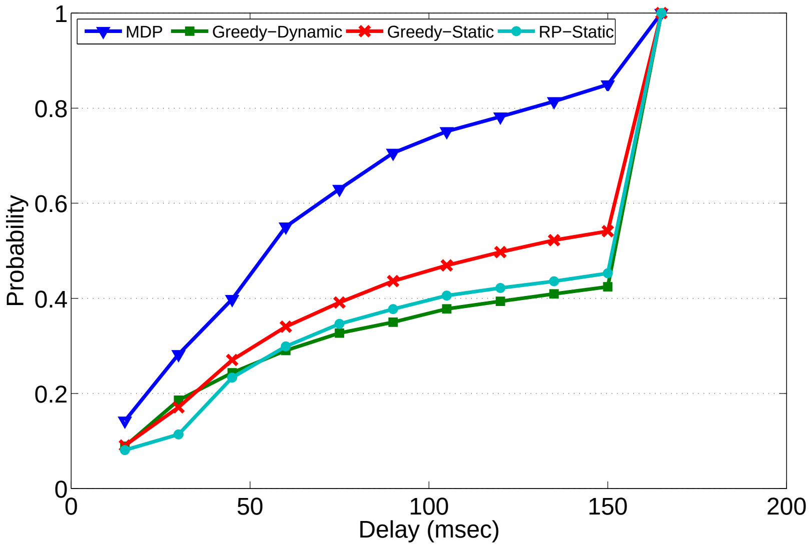

Figure 16. CDF of delay for Ethernet (T-Labs to TU Berlin)

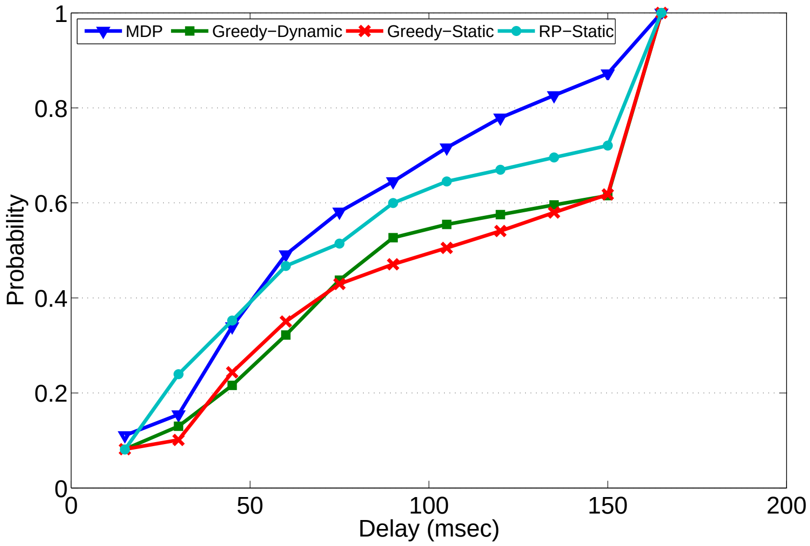

Figure 17. CDF of delay for 802.11g (T-Labs to TU Berlin)

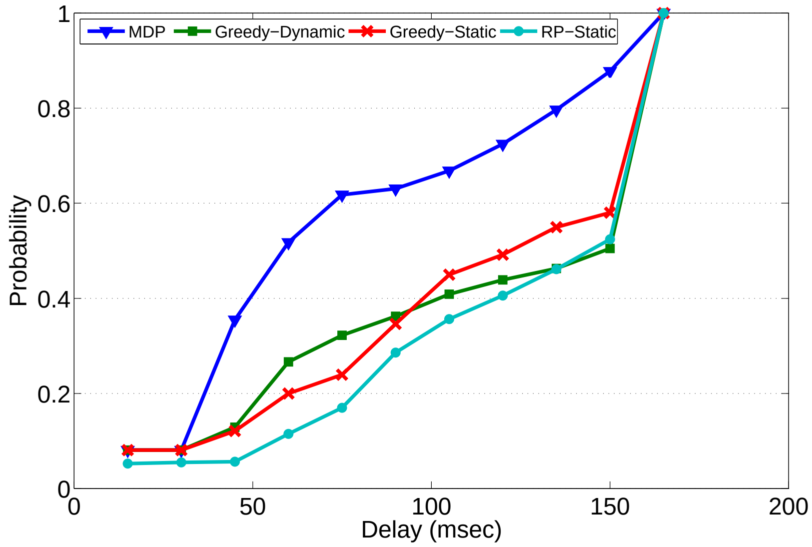

Figure 18. CDF of delay for 802.11b (T-Labs to TU Berlin)

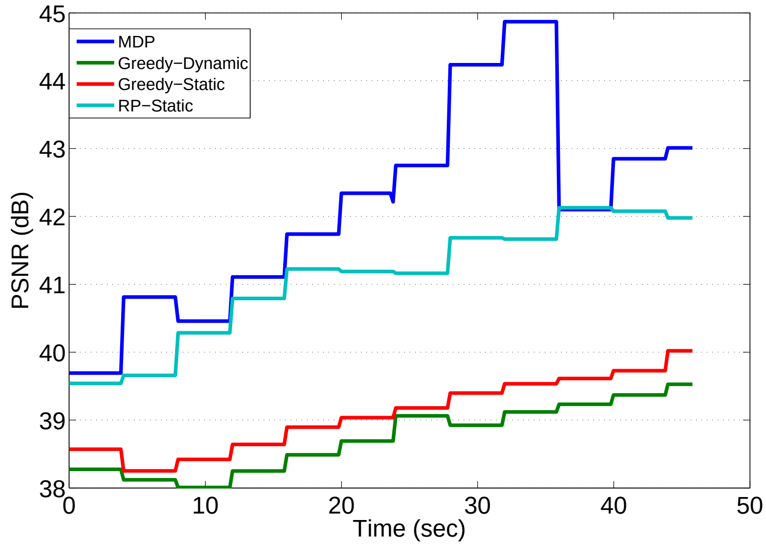

Figure 19. Average PSNR (T-Labs to TU Berlin)

Table VI. Packet loss rate (PLR) and flow rate statistics for TU Munich

	Greedy Static	RP Static	Greedy Dynamic	MDP
Avg. PLR(%)	35.95	24.65	32.8	16.34
Avg. ABR(Mbps)	2.5	4.61	4.57	6.54

Table VII. Impact of the Playout Deadline on PLRs for TU Munich case.

T(ms)	Greedy Static	RP Static	Greedy Dynamic	MDP
120	35.95	24.65	32.8	16.34
250	29.65	18.37	26.03	2.4
500	24.69	11.27	19.04	0

performance owing to the fact that network characteristics like ABR and delay vary due to fluctuations in cross-traffic and changes in the channel characteristics of wireless networks. The MDP based dynamic flow assignment policy presented in this work is able to efficiently utilize the diversity of available networks to enhance the QoS provisioning for applications. Through real world access network characteristic measurements and simulation experiments we demonstrate that our proposed flow assignment results in better performance in terms of packet delays and packet loss rates experienced by applications, and bandwidth utilization for different networks.

A noticeable aspect is the ability of MDP based flow assignment to offer low packet loss rates while allowing the flows to have their share of bit rates on different networks. Other static and dynamic policies are unable to keep the packet loss rate within acceptable limits. Hence, the MDP based flow assignment can easily guarantee acceptable PSNRs for multimedia flows whose performance depends on the bit rate and the packet loss. Again, as the deadlines for packet delivery becomes less stringent, the flow assignment policy results in a significant reduction in packet loss rates.

References

1. 'IEEE 802.21'. *http://www.ieee802.org/21/*.
2. 'The Network Simulator (ns-2)'. *http://www.isi.edu/ns/*.
3. Alpcan, T. and T. Başar: 2003, 'Global Stability Analysis of an End-to-End Congestion Control Scheme for General Topology Networks with Delay'. In:

Proc. of the 42nd IEEE Conference on Decision and Control. Maui, HI, pp. 1092 – 1097.

4. Alpcan, T. and T. Başar: 2005, 'A Utility-Based Congestion Control Scheme for Internet-Style Networks with Delay'. *IEEE Transactions on Networking* **13**(6), 1261–1274.
5. Alpcan, T., J. P. Singh, and T. Başar: 2007, 'A Robust Flow Control Framework for Heterogeneous Network Access'. In: *Proc.* 5^{th} *International Symposium on Modeling and Optimization in Mobile, Ad-hoc, and Wireless Networks (WiOpt)*.
6. Bertsekas, D.: 2001, *Dynamic Programming and Optimal Control*, Vol. 2. Belmont, MA: Athena Scientific, 2nd edition.
7. Chebrolu, K. and R. Rao: 2002, 'Communication using multiple wireless interfaces'. In: *Proc. IEEE Wireless Communications and Networking Conference (WCNC 2002)*, Vol. 1. pp. 327–331.
8. Cuevas, A., J. I. Moreno, P. Vidales, and H. Einsiedler: 2006, 'The IMS Platform: A Solution for Next Generation Network Operators to Be More Than Bit Pipes'. In: *IEEE Communications Magazine, issue on Advances of Service Platform Technologies*.
9. Fodor, G., A. Furuksar, and J. Lundsjo: 2004, 'On Access Selection Techniques in Always Best Connected Networks'. In: *Proc. ITC Specialist Seminar on Performance Evaluation of Wireless and Mobile Systems*.
10. Froyland, G.: 2001, *Nonlinear Dynamics and Statistics: Proceedings, Newton Institute, Cambridge, 1998*, Chapt. Extracting dynamical behaviour via Markov models, pp. 283–324. Birkhauser.
11. Furuksar, A. and J. Zander: 2005, 'Multiservice Allocation for Multiaccess Wireless Systems'. *IEEE Transactions on Wireless Communications* **4**, 174–184.
12. Guo, F., J. Chen, W. Li, and T. Chiueh: 2004, 'Experiences in Building a Multihoming Load Balancing System'. In: *Proc. IEEE INFOCOM*.
13. Konrad, A., B. Zhao, A. Joseph, and R. Ludwig: 2003, 'A Markov-Based Channel Model Algorithm for Wireless Networks'. *Wireless Networks* pp. 189–199.
14. Kumar, D., E. Altman, and J.-M. Kelif: 2007, 'Globally Optimal User-Network Association in an 802.11 WLAN and 3G UMTS Hybrid cell'. In: *Proc.* 20^{th} *International Telegraph Congress*. Ottawa, Canada.
15. Laboratory, U. N. R., 'CapProbe'. *http://www.cs.ucla.edu/NRL/CapProbe/*.
16. Mariz, D., I. Cananea, D. Sadok, and G. Fodor: 2006, 'Simulative Analysis of Access Selection Algorithms for Multi-access Networks'. In: *Proc. IEEE Internation Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2006)*.
17. Navratil, J. and R. L. Cottrell, 'Abing'. *http://www-iepm.slac.stanford.edu/tools/abing/*.
18. Salmatian, K. and S. Vaton: 2001, 'Hidden Markov Modeling for network communication channels'. In: *Proc. ACM SIGMETRICS*.
19. Shakkottai, S., E. Altman, and A. Kumar, 'The Case for Non-cooperative Multihoming of Users to Access Points in IEEE 802.11 WLANs'. In: *Proc. IEEE INFOCOM*.
20. Singh, J. P., T. Alpcan, X. Zhu, and P. Agrawal: 2007, 'Towards Heterogeneous Network Convergence: Policies and Middleware Architecture for Efficient Flow Assignment, Rate Allocation and Rate Control for Multimedia Applications'. In: *Workshop on Middleware for Next-generation Converged Networks and*

Application (MNCNA), ACM/IFIP/USENIX 8^{th} *International Middleware Conference*.

21. StuhlMuller, K., N. Farber, M. Link, and B. Girod: 2000, 'Analysis of video transmission over lossy channels'. *IEEE Journal on Selected Areas in Communication* **18**(6).
22. Thompson, N., G. He, and H. Luo: 2006, 'Flow Scheduling for End-host Multihoming'. In: *Proc. IEEE INFOCOM*.
23. Vidales, P., J. Baliosion, J. Serrat, G. Mapp, F. Stejano, and A. Hopper: 2005, 'Autonomic System for Mobility Support in 4G Networks'. *IEEE Journal on Selected Areas in Communications* **23**(12).
24. Yaiche, H., R. Mazumdar, and C. Rosenburg: 2000, 'A game theoretic framework for bandwidth allocation and pricing in broadband networks'. **8**(5), 667–678.
25. Zemlianov, A. and G. de Veciana: 2005, 'Cooperation and decision-making in a Wireless multi-provider setting'. In: *Proc. IEEE INFOCOM*. pp. 1–14.
26. Zhu, X., P. Agrawal, J. P. Singh, T. Alpcan, and B. Girod: 2007, 'Rate Allocation for Multi-user Video Streaming over Heterogeneous Access Networks'. In: *Proc. ACM Multimedia*.