An Optimal Flow Assignment Framework for Heterogeneous Network Access

Jatinder Pal Singh and Tansu Alpcan
Deutsche Telekom Laboratories
Ernst-Reuter Platz 7
10587 Germany
Email: {jatinder.singh, tansu.alpcan}@telekom.de

Piyush Agrawal and Varun Sharma
Department of Computer Science and Engineering
Indian Institute of Technology
Kanpur, India
Email: {piyushag, varunsh}@iitk.ac.in

***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 (lower packet delays and packet loss rates) for the flows, as compared to policies which do not perform dynamic flow assignment but statically allocate flows to different networks using heuristics like average available bit rate on the networks.**

I. 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 [1]. IEEE 802.21 [2] is delineating a framework to enable handovers and interoperability between heterogeneous wireless and wireline networks. The IP Multimedia Subsystems (IMS) [3] 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) [4] 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 [5] 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 the simulated framework 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.

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 [6–8]. 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 [9]. 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 [1]. The work, however, does not address efficient simultaneous use of heterogeneous networks and does not consider wireline settings. Similarly, the work [10] 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 [11], the authors have explored design of a network comprised of 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 [12]. 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 [13], and in [14] 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 [13], [14] 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.

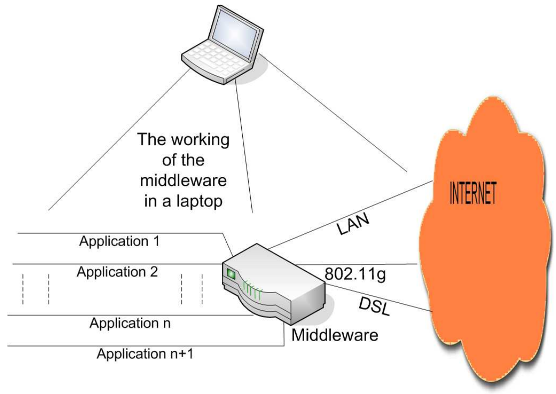

Fig. 1. Middleware functionality in a device

Flow scheduling for collaborative Internet access in residential areas via multihomed client devices is discussed in [15]. 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.

The rest of this paper is organized as follows: We present the system model and analytical framework for flow assignment in Section II. In Section III, we describe measurement tests demonstrating heterogeneity in network characteristics. The performance evaluation of the flow assignment framework is presented in Section IV. A discussion on results and flow assignment in heterogeneous networks is presented in Section V, and the paper is concluded in Section VI.

II. 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. 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 [16].

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 optimal policy is periodically evaluated for the assignment of each flow in the system to a suitable network.

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. We map the range of delay and ABR values of each network to a set of quantized states. Let $S_d^i := \{s_{d1}^i, \ldots, s_{dM}^i\}$ and $S_r^i := \{s_{r1}^i, \ldots, s_{rM}^i\}$ 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 $S := \prod_{i\in} S^i$ 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 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 [17, 18]. In addition, there exists well-established computational and theoretical methods to optimize Markovian processes [4]. 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

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

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

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

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 [19]. 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

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

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 SXS. Note that u assumes a values 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. and can then be expressed as :

$$R(s, u) = \beta \hat{R}^u(r^u, d^u) + (1-\beta) \sum_{i \in I \setminus \{u\}} \hat{R}^i(r^i, d^i), \quad (1)$$

where $\hat{R}^j(r^j, d^j)$ is the reward offered by network j to a flow in terms of the ABR r^j and delay d^j of the network. $\beta \in [0, 1]$ represents the tradeoff between selfishness and global good due to an action. The first term in (1) is the reward gained by a flow as a consequence of the action u directing it to a specific network in I. The second term represents the aggregate reward functions in state s for the other networks. For the value of β as 1, the MDP when executed for the assignment of a flow will only account for the reward received by the flow on the assigned network while evaluating the optimal allocation policy. On the other hand when β is 0, the MDP while evaluating the the optimal policy will only take into consideration the rewards of the networks other than the one to which a flow is assigned.

We formulate $\hat{R}^j(r^j, d^j)$ in (1) to represent the reward offered by a network to flows which exhibit the characteristics of video traffic:

$$\hat{R}^j(r^j, d^j) = f(r^j)\ u_s(r^j - r_{min})\ u_s(d^j - T), \quad (2)$$

where $f(r^j)$ is a concave utility function, and $u_s(r^j)$ and $u_s(d^j)$ are unit step functions. The flows represented in (2) 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 $f(r^j)$ in (2), we employ video encoder rate distortion models from [20] and adopt the following form:

$$f(r^j) = 10log_{10}(255^2/D(r^j)), \quad (3)$$

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

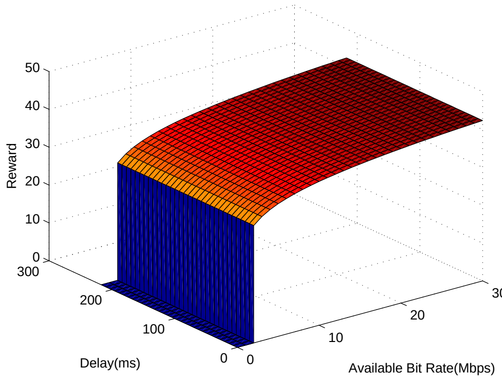

Fig. 2. Reward (with units of PSNR (dB)) offered by a network to an incoming flow for $\theta = 97.8$, $r_0 = 0.075$ (in Mbps), $D_0 = 0.49$, $T = 150ms$, and $R_{min} = 2Mbps$

by

$$D(r^j) = \frac{\theta}{r^j - r_0} + D_0. \quad (4)$$

The parameters θ, D_0, and r_0 can be estimated [20] from empirical rate-distortion curves via regression techniques. $\hat{R}^j(r^j, d^j)$ is plotted in Fig. 2 for a representative scenario.

Now the reward per stage in the MDP framework, $g : S \times I \longrightarrow \Re$, under a stationary policy $\mu(s) : \; S \longrightarrow I$ 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 :

$$J_\mu(s) := \lim_{N\to\infty} \sum_{n=0}^{N} \alpha^n g(s(n), \mu(s)), \quad (5)$$

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

$$J^*(s) := \max_{\mu\in\Pi} J_\mu(s), \quad s \in S, \quad (6)$$

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 ([4], Chapter 1) there exists an optimal deterministic stationary policy $\mu^*(s)$ that solves the *Bellman's equation*, i.e.,

$$J_{\mu^*}(s) = g(s(n), \mu^*(s)) + \alpha J_{\mu^*}(s\, M(\mu^*(s))), \quad \forall s \in S. \quad (7)$$

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 (7) actually corresponds to a single iteration or recursion of the venerable dynamic programming algorithm (DPA)

$$J_{n+1}(s) = \max_u g(s(n), u) + \alpha J_n(s(n)M(u)), \quad (8)$$

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

$$J^* = \lim_{N\to\infty} J_{n+1}(s), \quad (9)$$

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 [4].

The value iteration algorithm as described above is executed and updated periodically by the middleware on a device to evaluate the optimal control policy $\mu^*(s)$ for the assignment of a flow to an access network. The state space S is ascertained from the ABR and RTT measurements made by the middleware to the destination of a flow. 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.

The middleware gives the flows on a device a token in a round robin fashion and the flow with the token seeks middleware consultation and is reassigned according to the control policy. Whenever a flow is reassigned to a different network that it was previously on, it is provided a rate which is half of the ABR on that network. On the other hand, if the reassignment decision for a flow is to have it stay on the same network, the rate of the flow is increased by half of ABR on the network.

We compare the MDP based flow assignment policy described above with a static flow assignment with Additive Increase Multiplicative Decrease (AIMD) rate control policy which allocates newly admitted flows on suitable networks and does not reassign them for the rest of their lifetime. The flows are admitted with a rate equal to the r_{min} for their class. Thereafter the flows exercise an AIMD policy for rate adaptation. Each flow increases its rate by Δ_r every Δ_t seconds unless network congestion is perceived by a flow in which case it drops its rate by $(r - r_{min})/2$. 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 (2)) to a flow upon its admission. We term this policy as greedy-AIMD. Under the other static assignment scheme, flows are allocated to different networks in proportion to the average ABR on the networks. We call this assignment scheme Rate Proportional AIMD (RP-AIMD).

III. Network measurements and modeling

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 [21] 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 [22] 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.

Abing server is run at the machines at Stanford, TU Munich and TU Berlin, and the clients at machines in T-Labs in Berlin. 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.

TABLE I

AVAILABLE BIT RATE AND RTT FROM T-LABS TO STANFORD UNIVERSITY

		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

		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

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

TABLE III

AVAILABLE BIT RATE AND RTT FROM T-LABS TO TU BERLIN

		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

and their standard deviations to different destinations and for different networks for the 2 hour traces. Ethernet can be seen to 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 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 IV.

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.

IV. 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 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.

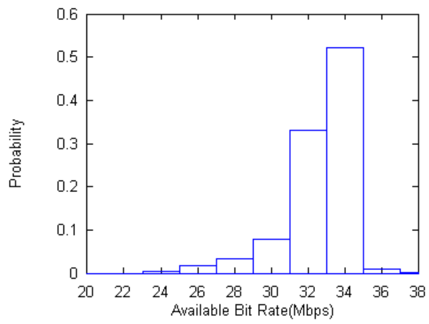

Fig. 3. Available bit rate on Ethernet from T-Labs to Stanford

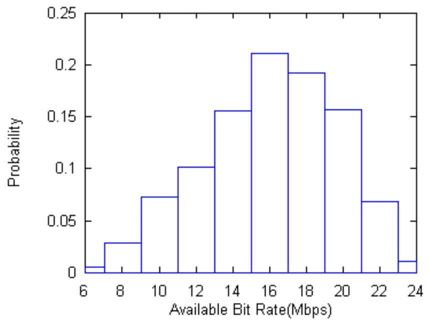

Fig. 4. Available bit rate on 802.11g from T-Labs to Stanford

These characteristics are obtained from the practical measurements performed in real networks settings - e.g. the ones described in Section III.

2) **Flow Assignment** : An instance of hash classifier [5] 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

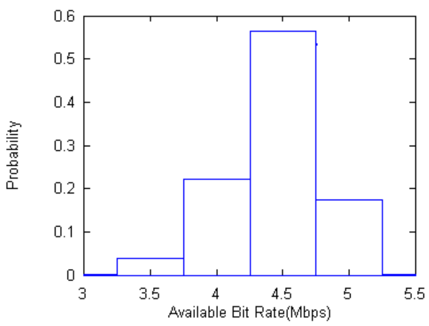

Fig. 5. Available bit rate on 802.11b from T-Labs to Stanford

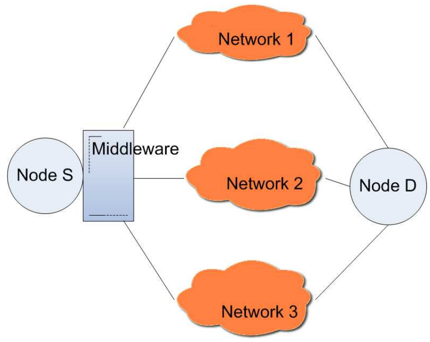

Fig. 6. Network topology in ns-2

from the Internet.

3) ***N*etwork delay measurement** : We employ CapProbe [23] 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 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 [5] 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 III. 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 II with $r_{min} = 2$ Mbps and $T = 150$ ms. At the beginning of a simulation, a total of 14 flows arrive with a rate of 2 Mbps each and an inter-arrival time of 0.5 seconds. 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-AIMD policy, a flow upon its arrival is allocated to a network that offers the maximum instantaneous reward given by (2). For RP-AIMD, 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 $(r - r_{min})/2$. This cooperative rate allocation scheme ensures efficient utilization of the network bit rates while preventing excessive rate fluctuations for the flows.

For the MDP based flow assignment, the parameter β value in (1) is selected as 0.8. We will discuss more about the selection of β in Section V. 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.

For evaluating the MDP policy, the middleware needs information about the system state transitions via-a-vis the control actions. For the purpose we maintain a sliding window of last 800 transitions and action tuples. The state transitions associated with an action are ascertained as those occurring from the state at the time of the control action to the states visited before the next action.

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 100 seconds 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 II 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 policies with dynamic MDP based approach. We first discuss the performance of flow assignment schemes for T-Labs to Stanford hosts. While the MDP based approach dynamically assigns the flows to different networks during the simulation, the greedy-AIMD policy leads to an allocation of 11, 3, and 0 flows on the Ethernet, 802.11g and 802.11b respectively and the RP-AIMD leads to a distribution of 9, 4, and 1 flows on the networks.

Fig. 7 shows the average rate of flows during the simulation run for different flow assignment schemes. Figs 8, 9, and 10 respectively plot the ABR for different networks in a window from 1000 to 1500 seconds of simulation run. As apparent from Fig 8, the bandwidth on the IEEE 802.11b network for the greedy-AIMD remains unutilized as no flow is assigned to the network. The packet loss rate during a simulation run for different flow assignment schemes is compared in Fig. 11.

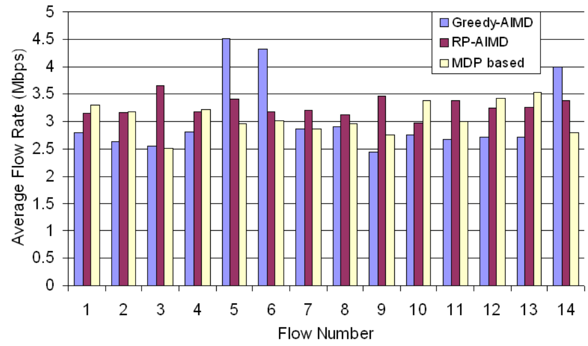

Fig. 7. Average flow rate for T-Labs to Stanford case

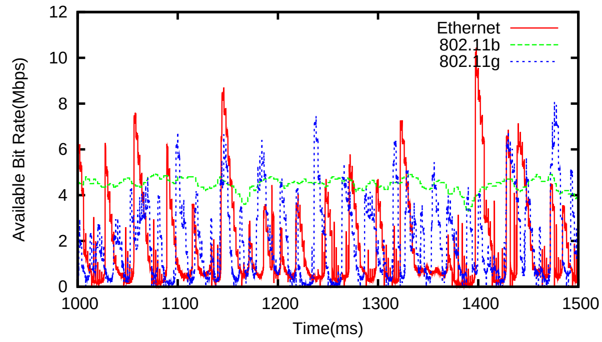

Fig. 8. Available bit rate for greedy-AIMD flow assignment for T-Labs to Stanford case

Tables IV shows the average of packet loss rate and flows rate for the 14 flows. The greedy-AIMD policy results in a higher average flow rate and hence suffers a high packet loss rate. As can be seen in Fig. 11 and Table IV, the MDP based flow assignment policy leads to a reduction of packet loss rate from around 45% for Greedy-AIMD and 40% for RP-AIMD to a significantly lower 2%.

TABLE IV

PACKET LOSS RATE (PLR) AND FLOW RATE STATISTICS FOR STANFORD UNIVERSITY

	Greedy AIMD	RP AIMD	MDP
Avg. PLR(%)	43.69	38.74	1.78
Avg. ABR(Mbps)	2.77	1.76	2.7

The MDP based flow assignment leads to significantly lower delays on all three networks. The cdfs of the delays (truncated at 150 ms, the playout deadline for the flows) for greedy-RP and MDP based flow assignment are plotted in Figs 12, 13, and 14. For greedy-AIMD, the delay performance is further worse than RP-AIMD on Ethernet and 802.11g networks; however delays are low for this scheme on 802.11b since no flows are allocated to the network.

We next discuss the performance for T-Labs to TU Munich and TU Berlin cases. Since Ethernet for TU Munich and TU Berlin cases has high ABRs (Tables II, III) the greedy-AIMD policy leads to assignment of all flows to Ethernet as

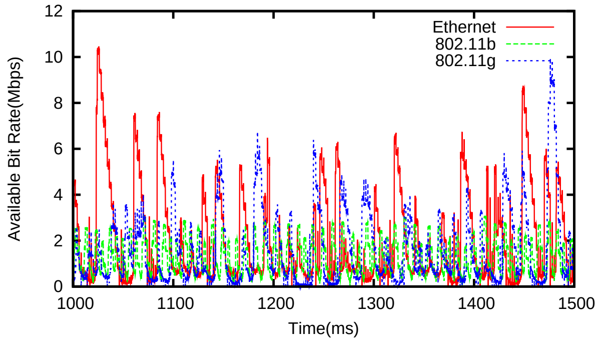

Fig. 9. Available bit rate for RP-AIMD flow assignment for T-Labs to Stanford case

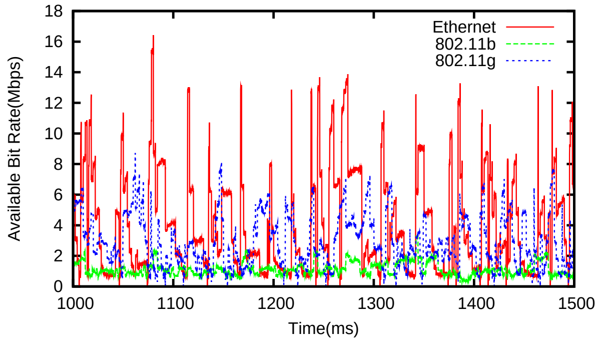

Fig. 10. Available bit rate for MDP based flow assignment for T-Labs to Stanford case

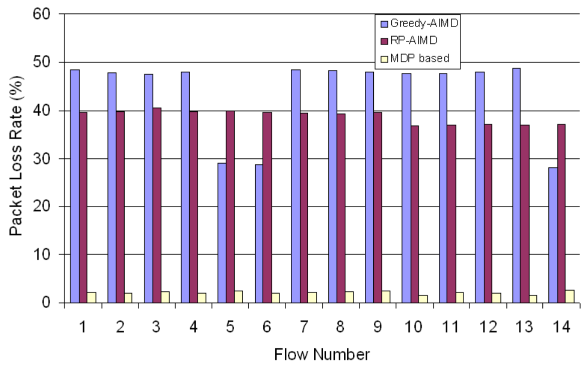

Fig. 11. Average percentage packet loss for T-Labs to Stanford case

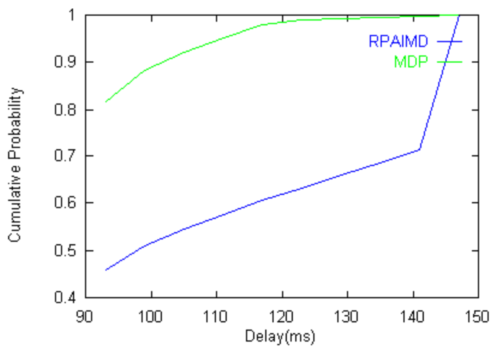

Fig. 12. CDF of delay for Ethernet for T-Labs to Stanford case

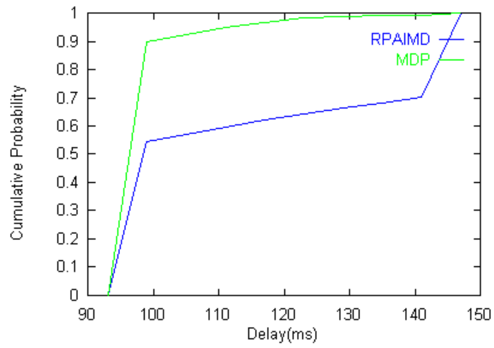

Fig. 13. CDF of delay for 802.11g for T-Labs to Stanford case

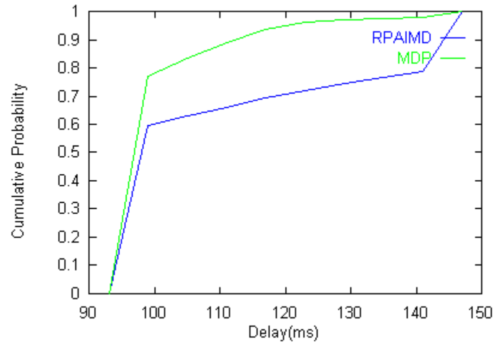

Fig. 14. CDF of delay for 802.11b for T-Labs to Stanford case

they arrive. The RP-AIMD results in a respective distribution of 12,2,0 and 11,2,1 flows on Ethernet, 802.11g, 802.11b networks for TU Munich and TU Berlin cases. As before, the MDP based approach dynamically assigns flows to networks during the simulation runs.

The packet loss rates for T-Labs to TU Munich and TU Berlin cases are shown in Figs 15 and 16. Tables V and VI and shows the average packet loss rate and flows rate for the 14 flows for the destinations TU Munich and TU Berlin. While the greedy-AIMD policy suffers from high packet loss rates for all cases, MDP based flow assignment leads to the significantly lower packet loss rates. Again, the delay performance of the MDP based flow assignment scheme is better than the static policies and; the plots are suppressed for brevity.

TABLE V

PACKET LOSS RATE (PLR) AND FLOW RATE STATISTICS FOR TU MUNICH

	Greedy AIMD	RP AIMD	MDP
Avg. PLR(%)	34.21	30.56	11.88
Avg. ABR(Mbps)	8.87	5.21	5.19

TABLE VI

PACKET LOSS RATE (PLR) AND FLOW RATE STATISTICS FOR TU BERLIN

	Greedy AIMD	RP AIMD	MDP
Avg. PLR(%)	36.82	33.49	13.15
Avg. ABR(Mbps)	9.78	5.46	6.22

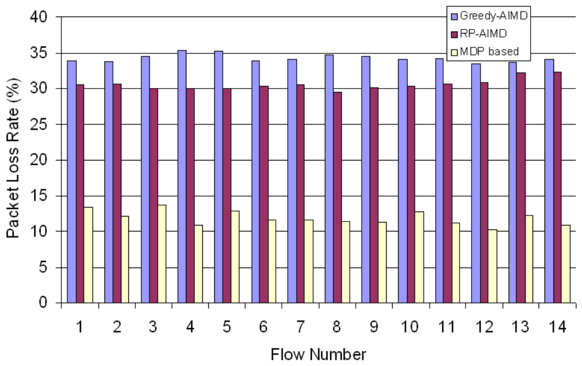

Fig. 15. Average percentage packet loss for T-Labs to TU Munich case

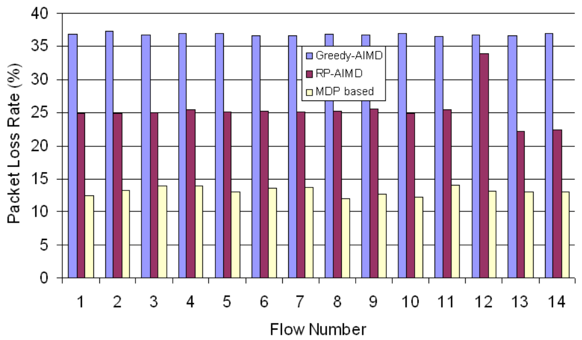

Fig. 16. Average percentage packet loss for T-Labs to TU Berlin case

The impact of variation of playout deadline (T) on the packet loss rate performance for the TU Berlin case 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-AIMD and RP-AIMD policies.

TABLE VII
IMPACT OF THE PLAYOUT DEADLINE FOR TU BERLIN

T(ms)	Greedy AIMD	RP AIMD	MDP
150	36.82	33.49	13.15
250	31.25	25.38	3.55
500	22.00	20.48	0.26

V. DISCUSSION

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 a static policy which assign flows based on heuristics like average ABR on the networks. Even in conjunction with a suitable rate control scheme, a static flow allocation policy suffers from degraded performance owing to the fact that network characteristics like ABR and delay vary to fluctuations in cross-traffic and changes in the channel characteristics for wireless networks.

A dynamic flow assignment policy is able to utilize the diversity of available networks to enhance the QoS provisioning for applications. For instance when ABR on a network drops or the delay shoots up, flows on the network may be reassigned to another network which may be experiencing a better network quality. An MDP based dynamical flow assignment presented in this work is demonstrated to result in better performance in terms of packet delays and packet loss rates experienced by applications, and bandwidth utilization for different networks.

We observed a tradeoff between selfish and global good as represented by the value of β in (1). For low values of β representing a higher concern for characteristics of other interfaces than the one associated with the control action, the flows did not always drive the system to a state where they received a good reward on a network dictated by the control action, and this lead to bandwidth wastage. On the other hand, overtly selfish behavior (β=1) pushed the system into high delay states as the flows would eagerly choose a state that maximized their reward even if the state represented high delays and low ABR for the other interfaces.

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 policies (RP-AIMD and greedy-AIMD) 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 (Table VII).

VI. CONCLUSION

In a setting where devices have access to multiple networks, the distribution of traffic flows amongst different networks can enable better network utilization than single network use at a time. However, the variation in network characteristics like ABR and delay make the problem of flow assignment challenging. A static flow allocation policy can result in unsatisfactory performance due varying characteristics of networks. However, the adaptive assignment of flows to different access networks results in a much better performance in terms of packet losses, delays and allocated bit rates. Such adaptive flow reassignment can be done via stochastic characterization of the networks and adopting an MDP based approach to optimally assign flows to networks.

ACKNOWLEDGMENT

We thank Xiaoqing Zhu for discussions on video traffic modeling. Thanks to Anja Feldmann for useful discussions on the organization and presentation of the work. We are also thankful to Sven Kleemann for providing the infrastructure for network measurements. Many thanks to the anonymous reviewers for their useful inputs and suggestions on the presentation of the work.

References

[1] P. Vidales, J. Baliosion, J. Serrat, G. Mapp, F. Stejano, and A. Hopper, "Autonomic sytem for mobility support in 4G networks," in *IEEE Journal on Selcted Areas in Communications'05*, vol. 23, no. 12, Dec. 2005.

[2] IEEE 802.21. [Online]. Available: http://www.ieee802.org/21/

[3] A. Cuevas, J. I. Moreno, P. Vidales, and H. Einsiedler, "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*, Aug. 2006.

[4] D. Bertsekas, *Dynamic Programming and Optimal Control*, 2nd ed. Belmont, MA: Athena Scientific, 2001, vol. 2.

[5] The network simulator (ns-2). [Online]. Available: http://www.isi.edu/ns/

[6] H. Yaiche, R. Mazumdar, and C. Rosenburg, "A game theoretic framework for bandwidth allocation and pricing in broadband networks," in *IEEE/ACM Transaction on Networking*, vol. 8, no. 5, Oct. 2000, pp. 667–678.

[7] T. Alpcan and T. Başar, "A utility-based congestion control scheme for Internet-style networks with delay," *IEEE Transactions on Networking*, vol. 13, no. 6, pp. 1261–1274, December 2005.

[8] ——, "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, December 2003, pp. 1092 – 1097.

[9] K. Chebrolu and R. Rao, "Communication using multiple wireless interfaces," in *Proc. IEEE Wireless Communications and Networking Conference (WCNC 2002)*, vol. 1, 2002, pp. 327–331.

[10] G. Fodor, A. Furuksar, and J. Lundsjo, "On access selection techniques in always best connected networks," in *Proc. ITC Specialist Seminar on Performance Evaluation of Wireless and Mobile Systems*, Sept. 2004.

[11] A. Zemlianov and G. de Veciana, "Cooperation and decision-making in a wireless multi-provider setting," in *Proc. IEEE INFOCOM*, 2005, pp. 1–14.

[12] S. Shakkottai, E. Altman, and A. Kumar, "The case for non-cooperative multihoming of users to acess points in IEEE 802.11 WLANs," in *Proc. IEEE INFOCOM*, Barcelona, Spain, Apr. 2006.

[13] A. Furuksar and J. Zander, "Multiservice allocation for multiaccess wireless systems," *IEEE Transactions on Wireless Communications*, vol. 4, Jan. 2005.

[14] D. Mariz, I. Cananea, D. Sadok, and G. Fodor, "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)*, June 2006.

[15] N. Thompson, G. He, and H. Luo, "Flow scheduling for end-host multihoming," in *Proc. IEEE INFOCOM*, 2006.

[16] F. Guo, J. Chen, W. Li, and T. Chiueh, "Experiences in building a multihoming load balancing system," in *Proc. IEEE INFOCOM*, 2004.

[17] A. Konrad, B. Zhao, A. Joseph, and R. Ludwig, "A markov-based channel model algorithm for wireless networks," *Wireless Networks*, pp. 189–199, September 2003.

[18] K. Salmatian and S. Vaton, "Hidden markov modeling for network communication channels," in *Proc. ACM SIGMETRICS*, 2001.

[19] G. Froyland, *Nonlinear Dynamics and Statistics: Proceedings, Newton Institute, Cambridge, 1998*. Birkhauser, 2001, ch. Extracting dynamical behaviour via Markov models, pp. 283–324.

[20] K. StuhlMuller, N. Farber, M. Link, and B. Girod, "Analysis of video transmission over lossy channels," in *IEEE Journal on Selected Areas in Communication*, vol. 18, no. 6, June 2000.

[21] J. Navratil and R. L. Cottrell. Abing. [Online]. Available: http://www-iepm.slac.stanford.edu/tools/abing/

[22] ——. INCITE: Edge-based traffic processing and inference for high performance networks. [Online]. Available: www-iepm.slac.stanford.edu/about/status/incite/incite-progress-feb04.doc

[23] U. N. R. Laboratory, *CapProbe*. [Online]. Available: http://www.cs.ucla.edu/NRL/CapProbe/