Hindawi Publishing Corporation
Advances in Multimedia
Volume 2007, Article ID 70429, 7 pages
doi:10.1155/2007/70429

Research Article

Bandwidth Estimation in Wireless Lans for Multimedia Streaming Services

Heung Ki Lee,[1] Varrian Hall,[1] Ki Hwan Yum,[2] Kyoung Ill Kim,[3] and Eun Jung Kim[1]

[1] *Department of Computer Science, Texas A&M University, College Station, TX 77843-3112, USA*
[2] *Department of Computer Science, The University of Texas at San Antonio, San Antonio, TX 78249-1644, USA*
[3] *Electronics and Telecommunications Research Institute (ETRI), 161 Gajeong-Dong, Yusung-gu, Daejeon 305-700, South Korea*

Received 2 November 2006; Accepted 21 December 2006

Recommended by Jianfei Cai

The popularity of multimedia streaming services via wireless networks presents major challenges in the management of network bandwidth. One challenge is to quickly and precisely estimate the available bandwidth for the decision of streaming rates of layered and scalable multimedia services. Previous studies based on wired networks are too burdensome to be applied to multimedia applications in wireless networks. In this paper, a new method, *IdleGap*, is suggested to estimate the available bandwidth of a wireless LAN based on the information from a low layer in the protocol stack. We use a network simulation tool, NS-2, to evaluate our new method with various ranges of cross-traffic and observation times. Our simulation results show that *IdleGap* accurately estimates the available bandwidth for all ranges of cross-traffic (100 Kbps $\sim$ 1 Mbps) with a very short observation time of 10 seconds.

1. INTRODUCTION

Since introduced commercially in 1995, multimedia streaming services have become one of the most promising internet services currently available. In addition, wireless local area networks (WLANs) make multimedia streams commonplace, and terminals are diversifying into hand-held devices such as PDAs, laptops, and audio/video players. These heterogeneous devices have different access patterns and mobility [1]. Most multimedia streams are hungry for stable network bandwidth, but a shared medium WLAN may not support it. To meet their bandwidth requirements, rate scalability can be achieved by layered video representation [2, 3]. However, there are still problems in estimating the point in time to change the bit rate of the transmitted bit stream. Estimating the available network bandwidth in a WLAN is very challenging and crucial for multimedia streaming services.

Although there can be various wireless environments where multimedia services are provided, we mainly focus on the WLAN shown in Figure 1. In this figure, an Internet-based set top box (STB) is the interface between a wired network and a wireless network. Even though wired networks can provide high and stable bandwidths, fragile wireless networks may not support them. Therefore, for layered streaming services, it is very critical for the STB to know the available wireless network bandwidth.

In a wireless network, the IEEE 802.11 protocol in distributed coordination function (DCF) mode, based on CSMA/CA algorithm, is becoming a de facto standard. Previous studies [4–6] based on the bandwidth estimation of wired environments are not applicable to wireless networks that use the DCF protocol. Multimedia streaming is a soft real-time service where each frame is delay-sensitive. Swiftness and availability are critical for real-time system. During bandwidth deviations, the rate of the transmitted multimedia streams should change expeditiously. The accuracy of previous works, *Spruce* [4] and *ProbeGap* [6], is dependent on probing time and the volume of the packets for probing. *ProbeGap* produces good estimates at low cross-traffic rates (2 Mbps cross-traffic regardless of the cross-traffic packet size); however, it significantly overestimates available bandwidth when the cross-traffic is high (4 Mbps cross-traffic generated with 300-byte packets) [6]. Influence by cross-traffic on probe packet sequences causes probe packets in sequences to be split up or even lost.

Our contribution in this paper is twofold. First, we suggest *IdleGap*, which is a bandwidth estimation tool for a real-time system in a wireless network. Second, our system

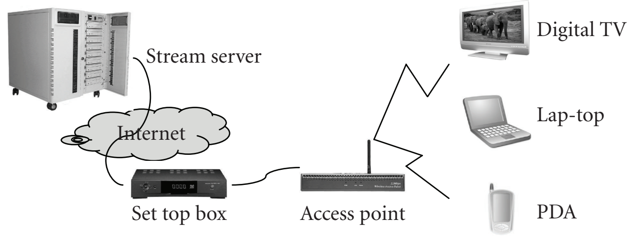

Figure 1: Stream service based on the STB and 802.11.

is independent of cross-traffic. We estimate the available bandwidth via the ratio of free time in the wireless links. To get the ratio of idle time in a wireless network, information from network management at the low layer is used. It provides us with an efficient and fast method for estimating the available bandwidth. The rest of the paper is organized as follows. Section 2 shows the related work in estimating bandwidth and discusses the cross-layer. In Section 3, our new method, *IdleGap*, is proposed and known challenges in bandwidth estimation are addressed. After presenting the results of our method and other tools in Section 4, we conclude this paper in Section 5.

2. RELATED WORK

2.1. Bandwidth estimation in broadband networks

Since the introduction of *Cprobe* [7], many tools have been suggested. *Cprobe* [7] uses Internet Control Message Protocol (ICMP) packet trains to estimate the current congestion along a path. *Cprobe* bounces a short stream of echo packets of a target server and records the time between the receipt of the first packet and the receipt of the last packet. Dividing the number of bytes sent by this time yields a measure of available bandwidth. In order to tolerate packet drops and possible reordering of packets, *Cprobe* uses results of four separate 10-packet streams when calculating the available bandwidth. *Cprobe*'s successors, *Spruce* [4] and *IGI* [8], use the interval of consecutive probe packets, since the interval or gap between probe packets increases in heavy cross-traffic. *Spruce* and *IGI* are both designed based on the ProbeGap model [6] which assumes a single bottleneck. *Spruce* samples the arrival rate at the bottleneck queue before the first packet departs the queue. *Spruce* calculates the number of bytes received at the queue between two probes for the interprobe spacing at the receiver. *Spruce* then computes the available bandwidth as the difference between the path capacity and the arrival rate at the receiver bottleneck. The *IGI* [8] algorithm sends a sequence of packet trains with an increasing initial gap, from the source to the destination host. *IGI* monitors the difference between the average source (initial) and destination (output) gap and terminates when it becomes zero. *Topp* [9] and *Pathload* [5] are also based on the rate of incoming packets. The comparison of the incoming rate from the sender side to the outgoing rate at the receiver side reveals the incoming rate to be less than or equal to the available bandwidth of the probing link. In *ProbeGap* [6], the link's idle time is the milestone for bandwidth estimation of a wireless network; however, *ProbeGap* also must send several probe packets over a specific interval.

All the methods outlined above introduce additional traffic into the link and require a probing sequence time to send and process the probing packets. To account for lost probes, additional probes are sent requiring more processing and filtering out of bad estimates. As a result, most of these methods may not be applicable to certain applications requiring instant bandwidth estimates, and if the link is congested, many probes may not reach the destination. Specifically, strict time bounds required of multimedia applications impose upper limits on delay and jitter in addition to the usual performance metrics of throughput and packet loss.

2.2. Cross-layer feedback

For efficient mobile device communication and interaction, cross-layer feedback is performed by a mobile device accessing its own protocol stack layers that contain information from the transmitted packets. Cross-layer feedback allows interaction between a layer and any other layers in the protocol stack. Packet information retrieval across the protocol stack layers, that is, cross-layering, provides very useful information about mobile devices in a wireless network. Several studies [7, 10–12] which have revealed interaction across-layers aid in improving a system. Shah et al. [10] proposes the use of a centralized *bandwidth manager* (BM), which obtains its channel time proportion (CTP) requirements from each flow at the start of its session. It uses this information to gauge what proportion of unit channel time each flow should be allotted. Its system takes advantage of cross-layer interaction between the application/middleware and the link layer. Davis [11] suggested an 802.11 management method that processes the captured frame to obtain the available bandwidth. The method describes a WLAN traffic probe that operates at the MAC layer and is capable of producing real-time information on resource usage on a per-station basis. For a QoS-sensitive application, a different priority at the MAC layer may be assigned based on the applications [12]. Carter and Crovella [7] used bandwidth probing to measure bandwidth and congestion at the application level. All these methods infer the ability to gather, compute, and share useful information for bandwidth estimation across the OSI layers. Eberle et al. [13] suggested a model for energy-efficient transmission that is based on cross-layer. They insert a *quality of energy manager* (QoEM) into the network protocol stack that manages the transmission.

2.3. Set top box

An STB is a device combining the functionality of analog cable converter boxes such as tuning and descrambling and computers such as navigation, interaction, and display. Today's STBs have four major components: a network interface, an MPEG decoder, graphics overlay, and a presentation engine [14].

2.3.1. STBs on the Internet

Recent successful deployments of IPTV over DSL in Europe and Asia have proven that telecommunication companies can successfully enter the market for television services. Last year Cisco acquired an STB manufacturer Scientific Atlanta (SA). Recently, another STB manufacturer, Motorola, agreed to purchase Kreatel, a Swedish manufacturer of IPTV STBs. For carrier networks and the digital home, this combination makes for a "triple play" solution integrating broadband video, voice, and data access into a single device.

The medium of delivery, the Internet, has also shown itself to be capable of delivering quality video and entertainment. As a result, the digital home consumer market has rapidly grown, and both Motorola and Cisco were aware of how the STB would play a key role in the digital home consumer market. The STB designers are being asked to support an array of new audio, video, and image formats as their products evolve into more open, networked devices. IPTV STBs may be enabled with the functions of personal video recorders (PVRs), digital media adapters (DMAs), voice over IP (VoIP), videophones, and more [15].

Due to the heterogeneous nature of home-based networked devices, each new device with additional functionality layers on different requirements. IPTV and VoD depend on streaming media over a wide area network (WAN) while media applications such as PVR and DMA add a media source in a home LAN environment. For IP video transmitted using the UDP protocol, packet loss can cause significant QoS reduction. A simple video stream can be severely degraded with low levels of packet losses, due to error propagation effects. Video quality is often represented in terms of peak signal-to-noise ratio (PSNR), which is a measure of the root mean square (RMS) error between the original and reconstructed video sequences.

Although all of the issues highlighted require a solution, it is a critical importance for the ability of the STB to adapt to the limited available bandwidth. Telchemy, a leader in VoIP and IPTV performance managements offers a lightweight software agent called VQmon/SA-VM that can be integrated into STBs [16]. VQmon/SA-VM transmits metrics back to service providers during video transmissions. The following are the feedback metrics.

(1) Video service transmission Quality (VSTQ) score, providing data on video transmission quality
(2) Video quality score (VQS), providing an estimate of user perceived quality.

Although this method provides a unique solution for the management of a service provider to STB transmissions, it does not provide a solution for an STB to end-user link management.

2.3.2. Our approach

Typically an STB receives a request from a client, retrieves the requested multimedia data from the server, and forwards it to the multimedia terminal. During this process, the STB can cache portions of the stream and forward the cached stream data to multimedia terminals through a shared resource, the wireless channel. The STB caches and forwards the streaming data between two different networks, wired and wireless networks, in order to reduce negative effects of network traffic such as late packets. The more resources assigned to handle the streams, the less jitter the terminal will experience within the network. The wireless channel is a limited shared resource available for servicing heterogeneous multimedia streams. Therefore, a simple and effective allocation strategy for the wireless channel is critical for improving the quality of the video streams delivered through the STB and the wireless network. In general, the streaming services with high quality may require more resources than the ones with low quality. Unfortunately, the estimation of the available resources required for each case has not been fully understood yet, so currently our research focuses on how to estimate the available resources for heterogeneous streaming services in this environment.

As shown earlier in Figure 1, an STB resides between the server and multimedia terminals, and relays the data flow from the server to the terminals and vice versa. Although the cost of the STB limits its functionality, a simple strategy implemented within the STB can improve the quality of multimedia services dramatically.

3. IDLEGAP USING NETWORK ALLOCATION VECTOR

3.1. Background

Bandwidth estimation is a prerequisite problem for real-time applications in wireless networks. There are two factors making this problem unique. First of all, unlike wired networks, traditional FIFO is not used to schedule bandwidth among connections in wireless networks. To avoid collisions in wireless networks, nodes are arranged in a distributed manner. This arrangement causes bandwidth estimation methods in wired networks using intervals [4, 8] or rates [5, 9] inapplicable for bandwidth estimation in wireless networks. Secondly, probing time for the available bandwidth should be short for time-sensitive multimedia streaming services. References [6, 11] suggested that idle time of a link in a wireless network can be a major milestone for estimating the available bandwidth as follows.

Let C be the capacity of the wireless network.[1] *Idle_rate* indicates the rate at which the link is idle. Then the available bandwidth (AB) can be obtained by the following product:

$$AB = C \times Idle\_rate. \tag{1}$$

However, previous methods [6, 11] using this formula cause too much overhead to be used in a real-time system for the estimation of the available bandwidth. In [6], too

[1] It can be changed by the negotiated data rate between a wireless node and the access point.

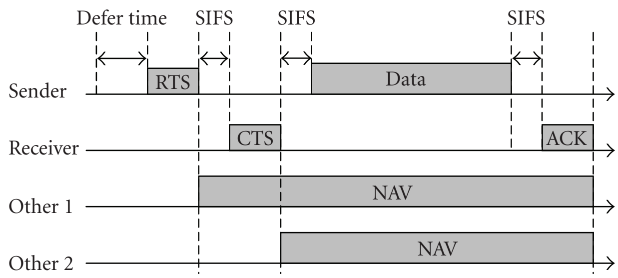

FIGURE 2: IEEE 802.11 DCF MAC protocol.

much time elapsed probing the link and analyzing probing data, and results showed multiple incorrect estimated values in heavy traffic. Reference [11] utilizes too much time in order to capture whole packets in the network and get node information from captured packets. For real-time applications such as multimedia streams, it is difficult to use these methods; therefore, in Section 3.2.2 we introduce an efficient method to calculate the *Idle_rate*.

3.2. IdleGap

3.2.1. Network allocation vector

When two nodes in a wireless network share the same access point (AP) but cannot hear each other, one node will not be able to know whether the other node is already using the shared resource, that is, the wireless channel. For addressing this hidden node problem, each node uses the network allocation vector (NAV) that shows how long other nodes allocate the link in the IEEE 802.11 DCF MAC protocol. Even though a node is located at a place where it cannot reach other active nodes, the node can know whether another node is already using the wireless network by checking its NAV. In Figure 2, when the sender sends RTS (request to send) to the receiver (AP), *Other-1* node that is reachable from sender updates its NAV. However, *Other-2* node does not update NAV, because it is not reachable from sender. When the receiver sends CTS (clear to send), *Other-2* node updates its NAV. The idle time in the wireless network can then be estimated from the NAV information.

3.2.2. Estimation of wireless link idle rate

All nodes in a WLAN share the same resource; that is, a wireless channel. If a node in a WLAN is utilizing the resource, the additional node(s) should await the release of the wireless channel. During a transmission in a WLAN, a node can be one of the following: sender, receiver, or onlooker. If a node transmits data to another node, it is a sender. A node is a receiver if receives data. Finally, when a node does not join the transmission, it is an onlooker.

The busy time of the link can be estimated by adding up all the transactions of nodes in the network as depicted in (2). Here T_l is the busy time of link l and $TT(i, j)$ indicates

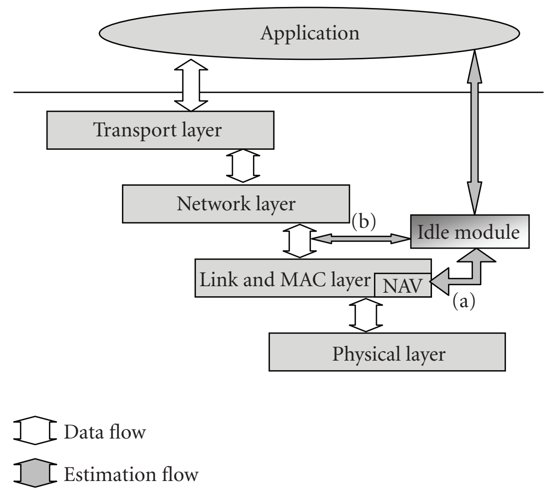

FIGURE 3: Architecture of idle module.

the transaction time between nodes i and j,

$$T_l = \frac{1}{2} \times \sum_{i=1}^{n} \sum_{j=1}^{n} (TT(i, j)). \tag{2}$$

Unfortunately, we cannot know all the transaction times from the nodes in the network. In addition, obtaining the transaction information can increase network traffic, hence affecting current traffic on the network. Therefore, we propose to obtain all the necessary information from one node in the network as follows.

The transaction time of node i can be obtained via the sum of the sending and receiving times to/from node i ($TT(i, j) = ST_i + RT_i$, where ST_i is the sending time from node i to j and RT_i is the receiving time from node j to i). For the transaction time between other nodes, we can get the onlooking time from the NAV in node i that is updated in other node transactions ($TT(i, j) = OT_i$, where OT_i is the onlooking time at node i). Therefore, we can estimate the busy time T_l in any node i in the network as shown in (3):

$$T_l = ST_i + RT_i + OT_i. \tag{3}$$

We can then obtain *Idle_rate* using the busy time:

$$Idle\_rate = 1 - \frac{\text{busy time}}{\text{total elapsed time}}. \tag{4}$$

3.2.3. System model

We propose to add an *idle module* in the MAC layer of a node in the network. This module obtains the busy time (T_l) from (a) and (b) in Figure 3. The transaction time of a node can be obtained through accessing outgoing and incoming packets ($ST_i + RT_i$) between the network layer and the link and MAC layer (shown in (b)). Idle module also gets the onlooking time (OT_i) from the NAV (shown in (a)). The updating process of the NAV triggers the *idle module* to update its value. An application can access the *idle module* to get the

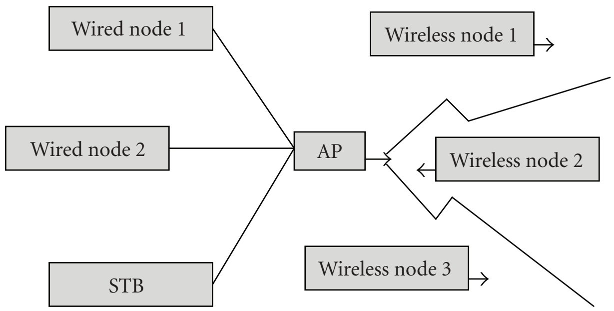

Figure 4: Simulation environment.

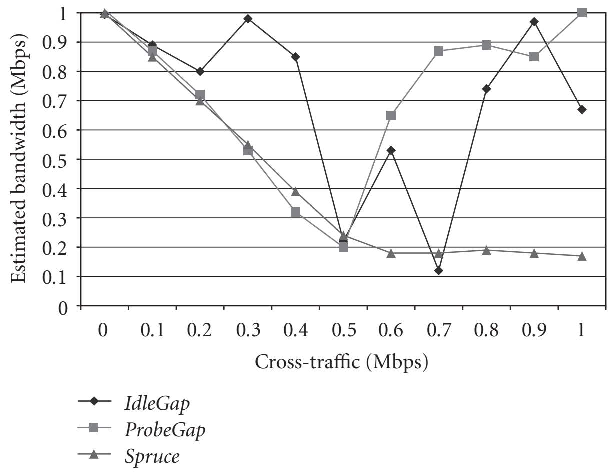

Figure 5: Estimated bandwidth with cross-traffic.

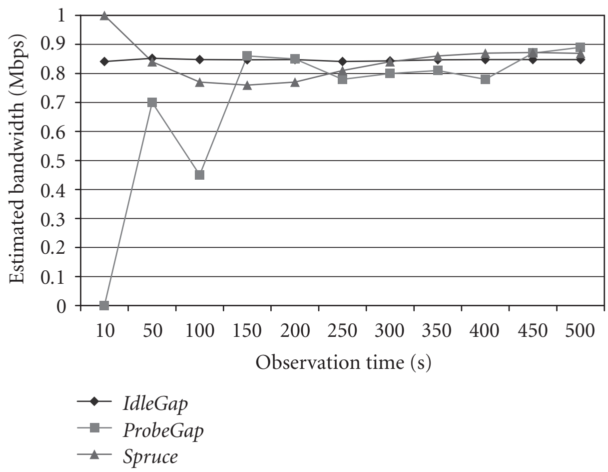

Figure 6: Estimated bandwidth with different observation times.

idle rate (1 − busy time/total elapsed time). Then applying the idle rate and link capacity C to (1), the estimated bandwidth of the link can be calculated with minimal effort. We call this method *IdleGap*.

4. EXPERIMENTAL RESULTS

To verify the performance of our *IdleGap* method, network simulations were conducted using NS-2. As shown in Figure 4, there are seven nodes including one STB, two other wired nodes, three wireless nodes, and an AP. In the wired network, the capacity of the link was set to 10 Mbps, while the capacity in wireless network was set to 1 Mbps.

In Figure 4, communication in the simulation via the AP involves three connections: wired node 1 to wireless node 2, wired node 2 to wireless node 1, and STB to wireless node 3. wired nodes 1 and 2 generate the cross-traffic, while the algorithm generates timestamps from packets received by STB via packets sent from wired node 3 to estimate the available bandwidth. We compare *IdleGap* with *ProbeGap* [6] and *Spruce* [4], which provides more accurate estimation than other previous works.

4.1. Experiments with increasing cross-traffic

Figure 5 shows the estimated available bandwidth value for each algorithm. The capacity of the wireless network in our simulation is 1 Mbps. Probing time for each algorithm is 1000 seconds and 200 probing packets are allowed. In light cross-traffic, *ProbeGap* produces bandwidth estimates reflective of measured available bandwidth values. However, it shows multiple transition points over 200 Kbps cross-traffic. In the original *Spruce* paper, the intrapair gap is set to the transmission time of the narrow link [4]. This causes the underestimation of the available bandwidth for the link. Therefore, the intrapair gap was calibrated to reflect the available 1.0 Mbps with no cross-traffic. Even after the calibration, *Spruce* overestimates the bandwidth severely with more than 0.5 Mbps cross-traffic. The reason is due to high drop rates with heavy cross-traffic. Thus, the estimated bandwidth value becomes polluted. This could cause the overestimation of the available bandwidth.

The *IdleGap*, which uses NAV to estimate bandwidth, shows the closest match to the real bandwidth. Note that after 0.6 Mbps cross-traffic, saturation occurs due to the overhead of the wireless network such as defer time and RTS/CTS.

4.2. Experiments with different observation times

In this experiment, we vary the observation time to estimate the available bandwidth. Since we focus on the effect of observation period, the cross-traffic is set to 10 Kbps, where all three schemes are able to estimate the bandwidth accurately as shown in Figure 5. *ProbeGap* and *Spruce* send the probes at intervals of 5 seconds [6]. Figure 6 shows the estimated values of the available bandwidth for *ProbeGap, Spruce,* and *IdleGap* between observation periods of 10 and 500 seconds. Until 250 seconds, *ProbeGap* and *Spruce* record values not reflective of measured available bandwidth. After 250 seconds, *ProbeGap* and *Spruce* values are near the measured bandwidth values. However, *IdleGap* generates values reflective of measured bandwidth for all periods. Therefore, we can conclude that *IdleGap* provides accurate estimations with short observation times.

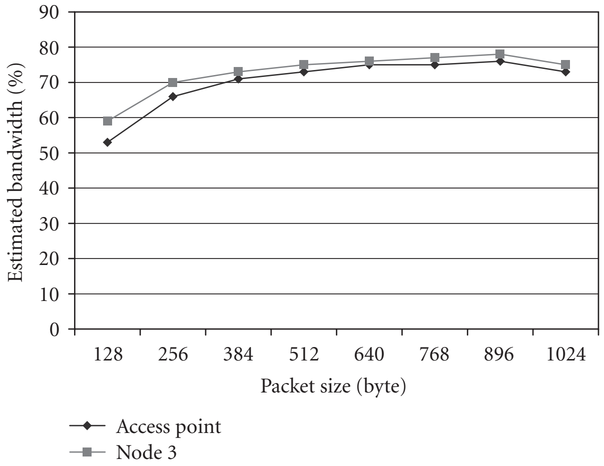

Figure 7: Estimated bandwidth with different packet sizes.

4.3. Experiments with different packet size

In Figure 7, the estimated idle times in the AP and node 3 are depicted with different packet sizes of the same cross-traffic. Cross-traffic in the simulation is 100 Kbits/sec (10%) and the packet size is changed from 128 to 1024 bytes. We observe that packet sizes between 512 and 896 bytes provide more accurate estimation. The estimated idle time with the small size packet is smaller than the one with the large size packet. It is because the overhead of the small packet transmission is larger than the one of the large packet transmission. In order to transmit a packet, the sender should send the RTS, CTS, and ACK to the receiver. The frequent transmission of small packets increases this overhead. That is why the *IdleGap* underestimates the available bandwidth with small size packets. On the other hand, with the largest size packets (1024 bytes), the estimated idle time is also decreased slightly. During the transmission, the large packet is broken into several fragments in the Mac layer to reduce the error rate, which again causes overhead.

Estimated bandwidth in the AP inclines to be smaller than the one in node 3. If node 3 is a hidden node, it receives only the CTS, not the RTS. Then, node 3 cannot detect the busy time gap between the RTS and the CTS.

5. CONCLUDING REMARKS

The most challenging aspect of multidynamic server selection media streaming services is the adaptive bit rate of each multimedia stream according to the network status; therefore, in this paper we focused on a method to estimate the available bandwidth of a wireless link. The method must have the following characteristics: (a) it should be applicable to real-time applications such as multimedia streaming services; (b) be simple and effective in estimating the available bandwidth, and (c) incur low overhead.

We have presented a new bandwidth estimation method, *IdleGap*, which can efficiently calculate the available bandwidth using the information collected from one node in a wireless network. *IdleGap* is simple and does not incur extra network overhead. The simulation result shows that *IdleGap* outperforms the other probing and bandwidth estimation methods, *ProbeGap* and *Spruce*.

REFERENCES

[1] T. Henderson, D. Kotz, and I. Abyzov, “The changing usage of a mature campus-wide wireless network,” in *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking (MOBICOM ’04)*, pp. 187–201, Philadelphia, Pa, USA, September-October 2004.

[2] J. Shin, J. W. Kim, and C.-C. J. Kuo, “Quality-of-service mapping mechanism for packet video in differentiated services network,” *IEEE Transactions on Multimedia*, vol. 3, no. 2, pp. 219–231, 2001.

[3] D. Quaglia and J. C. De Martin, “Delivery of MPEG video streams with constant perceptual quality of service,” in *Proceedings of IEEE International Conference on Multimedia and Expo (ICME ’02)*, vol. 2, pp. 85–88, Lausanne, Switzerland, August 2002.

[4] J. Strauss, D. Katabi, and F. Kaashoek, “A measurement study of available bandwidth estimation tools,” in *Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC ’03)*, pp. 39–44, Miami Beach, Fla, USA, October 2003.

[5] M. Jain and C. Dovrolis, “Pathload: a measurement tool for end-to-end available bandwidth,” in *Proceedings of Passive and Active Measurements Workshop (PAM ’02)*, pp. 14–25, Fort Collins, Colo, USA, March 2002.

[6] K. Lakshminarayanan, V. N. Padmanabhan, and J. Padhye, “Bandwidth estimation in broadband access networks,” in *Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC ’04)*, pp. 314–321, Taormina, Italy, October 2004.

[7] R. L. Carter and M. E. Crovella, “Dynamic server selection using bandwidth probing in wide-area networks,” Tech. Rep. TR-96-007, Computer Science Department, Boston University, Boston, Mass, USA, 1996.

[8] N. Hu and P. Steenkiste, “Evaluation and characterization of available bandwidth probing techniques,” *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 6, pp. 879–894, 2003, special issue in Internet and WWW Measurement, Mapping, and Modeling.

[9] B. Melander, M. Bjorkman, and P. Gunningberg, “A new end-to-end probing and analysis method for estimating bandwidth bottlenecks,” in *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM ’00)*, vol. 1, pp. 415–420, San Francisco, Calif, USA, November-December 2000.

[10] S. H. Shah, K. Chen, and K. Nahrstedt, “Dynamic bandwidth management in single-hop ad hoc wireless networks,” *Mobile Networks and Applications*, vol. 10, no. 1, pp. 199–217, 2005.

[11] M. Davis, “A wireless traffic probe for radio resource management and QoS provisioning in IEEE 802.11 WLANs,” in *Proceedings of the 7th ACM Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (ACM MSWiM ’04)*, pp. 234–243, Venezia, Italy, October 2004.

[12] G. Xylomenos and G. C. Polyzos, “Quality of service support over multi-service wireless Internet links,” *Computer Networks*, vol. 37, no. 5, pp. 601–615, 2001.

[13] W. Eberle, B. Bougard, S. Pollin, and F. Catthoor, "From myth to methodology: cross-layer design for energy-efficient wireless communication," in *Proceedings of the 42nd Design Automation Conference (DAC '05)*, pp. 303–308, Anaheim, Calif, USA, June 2005.

[14] A. Laursen, J. Olkin, and M. Porter, "Oracle media server: providing consumer based interactive access to multimedia data," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, vol. 23, no. 2, pp. 470–477, Minneapolis, Minn, USA, May 1994.

[15] S. Robertson and R. Rivin, "Analog Devices: Designing IPTV Set-Top Boxes Without Getting Boxed In," http://www.analog.com/.

[16] "Application Note IPTV Performance IPTV Performance Management," January 2005, http://www.telchemy.com/appnotes/Managing_IPTV_Service_Quality.pdf.