Active Techniques for Available Bandwidth Estimation: Comparison and Application

Alessio Botta[1], Alan Davy[2], Brian Meskill[2], and Giuseppe Aceto[1]

[1] Waterford Institute of Technology, Ireland, {adavy,bmeskill}@tssg.org
[2] University of Napoli Federico II, Italy, {a.botta,giuseppe.aceto}@unina.it

Abstract. There are various parameters for analyzing the quality of network communication links and paths, one attracting particular attention is available bandwidth. In this chapter we describe a platform for the available bandwidth estimation, a comparison of different tools for the estimation of this parameter, and an application of such estimation in a real-world application. In details, we describe a novel platform called UANM, capable of properly choosing, configuring, and using different available bandwidth tools and techniques in an autonomic fashion. Moreover, thanks to UANM, we show the results of a comparison of the performance of several tools in terms of accuracy, probing time and intrusiveness. Finally, we show a practical example of the use of the available bandwidth measurement: we describe an approach for server selection and admission control in a content distribution network based on the available bandwidth estimation.

Keywords: Available Bandwidth, Quality of Service, Server Selection

1 Introduction

In recent years, there has been an increased focus on measuring the quality of communication links over data networks [9, 8]. These communication links can represent logical or physical connections between two entities at various layers in the protocol stack, and over varying heterogeneous networking technologies. With the rise of application level traffic optimization within various application domains such as multimedia streaming [23] and content delivery server selection [21, 22], the assessment of the available bandwidth of communication links is invaluable. This chapter focuses on the estimation of the available bandwidth, a platform used for comparison between various available bandwidth estimation tools, and finally a usage scenario focusing on server selection. When looking beyond a single link and at the complete end-to-end path of a communication network, the measurement of available bandwidth becomes imperative [5]. This is generally termed the residual bandwidth available on the end-to-end path before additional congestion occurs. This measurement is useful for a variety of applications such as server selection, peer node selection, and video streaming.

There is a wide set of tools available for estimating this metric, each tailored to specific scenarios and requiring expert operators to obtain accurate results. In this paper we describe a novel platform called UANM, which has been designed to avoid the effect of the interference among concurrent measurement processes, and to automatically select and configure a measurement technique, according to the scenario. Moreover, offering a common generic model for the measurement tools (the *plugin API*) UANM allows for a fair comparison of the different techniques in terms of probing time, intrusiveness, and accuracy in each given scenario. In this paper, we show the results of a comparative analysis of different available bandwidth tools and techniques performed through UANM under varying network scenarios. Moreover, to show a use-case scenario for the Available Bandwidth estimation, we present an admission control and server selection framework. This framework relies on the measurement of available bandwidth between the client and the server to both choose a suitable server to deliver content and also assess whether adequate bandwidth is available to serve the request. We demonstrate that with appropriately configured available bandwidth measurements, improved control of end-to-end traffic flows over an unmanaged network infrastructure is possible.

The chapter is structured as follows: Section 2 provides background details and related work on the concepts of available bandwidth. Section 3 discusses the UANM platform and a comparison of available bandwidth estimation tools on the UANM platform. Section 4 demonstrates the application of available bandwidth estimation to admission control of video content over an unmanaged network. Finally, Section 5 concludes the chapter with a summary and future research challenges in the area of quality analysis of communication links.

2 Related Work

Available Bandwidth of a network path is defined as its remaining capacity, that is, the amount of traffic that can be sent along the path without congesting it [10]. Many tools exist for calculating it in an end to end context; that is, without any information from the intermediate topology. As such, different available bandwidth estimation tools vary in accuracy at this estimation. The many tools that have been proposed in the literature are often broadly categorised into two approaches, which we briefly discuss via some well known examples.

The Probe Gap Model (PGM)[17] approach uses probe packet pairs or packet trains to determine the available bandwidth. It uses these pairs by noting the difference between their network entry time gap, and their network exit time gap. The difference in this time gap is the time the bottleneck link required to service any non probing traffic on the bottleneck hop and this time can be used, along with the link capacity, to calculate the available bandwidth of the bottleneck link[34]. The Probe Rate Model[14] approach sends a train of packets and utilizes the concept of self-induced congestion[35] to determine the available bandwidth of the network path. Each packet train is forwarded through the network at a particular rate. The rate increases until particular self-induced characteristics are observed from the packet train such as a diversion from the initial packet train transmission rate. This method can pinpoint the available bandwidth of the path without knowledge of the physical capacity of the network links involved. Each of these approaches has its pros and cons, and none of them is better than the others in all the possible application scenarios. For this reason, the scientific community needs tools and platforms for using what is already available at the best of its possibilities, and, to this aim, several research papers in literature have compared the performance of available bandwidth estimation tools. Most of the comparison works have been done when presenting a new available bandwidth estimation tool. For instance, when presenting Traceband [16], the authors compared its performance with that of Spruce and Pathload on a real network using different traffic patterns. Results show that Traceband is faster and less intrusive than the others, and it achieves the same accuracy of Pathload. On the other hand, there are also works in which the authors compared several well known tools, without presenting new ones. For example, Goldoni et al. [15] compared the accuracy, the intrusiveness, and the convergence time of nine of the most wide spread tools on a real testbed with 100Mbps links, and with both constant bit rate (CBR) and Poisson cross-traffic. Other works [31, 24] evaluated such performance on very high speed networks. Some works [5] have also shown that the combined use of different techniques can increase the estimation accuracy. Summarizing, a first issue to be tackled for a proper available bandwidth estimation is the choice of the right tool according to the scenario.

Another important issue for the available bandwidth estimation is the fact that most of the existing tools can provide accurate results only if properly used and calibrated. This issue has been revealed and analyzed by different works in literature. For example, in 2004 Paxon et al. [27] claimed that a better design stage of measurement experiments is of great importance to avoid frequent mistakes, mainly due to the imperfections of tools. The authors basically reported that a calibration is usually needed to detect and correct possible errors. Other works [3, 34] analyzed commonly used tools on real testbeds and reported several pitfalls in which they can typically end.

These two issues are among the main motivations that drove us to design and implement UANM, a platform that helps the user to obtain accurate, fast and non intrusive available bandwidth estimations, choosing and configuring the right tool for the operating scenario. Architectures similar to UANM, which aim at taking all the measurement-related variables into the proper account, have already been presented in literature, although not specifically designed for the available bandwidth measurement. NetQuest [33], for example, firstly designs the experiments in order to best fit to the current scenario, and then it builds a global view of the network status. However, differently from UANM, NetQuest is not mainly interested in obtaining the best performance from the available bandwidth estimation, since it is more oriented to a wider knowledge of the general network status. Wide-scale infrastructures for network measurement and experimentation such as GENI [13] and Planetlab [12] have been created in order to perform experiments on a global level and study Internet-scale phenomena. Being designed to interact with third party measurement tools, UANM can leverage the existing infrastructures to perform the requested measurements;

moreover, thanks to its *user API*, it can be easily integrated in such infrastructures as a (compound) measurement tool. Similarly to our approach, Sommers et al. [32] proposed YAZ, an architecture whose main goal is to calibrate the existing tools in order to obtain the best results from the measurements. Differently from UANM, YAZ does not support concurrent experiments, does not consider the network status, and does not support third-party tools.

3 Comparing available bandwidth estimation tools through UANM

UANM is a distributed platform for network measurement supporting different techniques and tools. Full compliance and open interaction with existing tools is retained while offering a fair comparison environment, mutual exclusion of concurrent measurements and automatic selection and calibration of the tools. We report a brief description of UANM. More details are reported in [2, 1].

3.1 UANM architecture

The components of the platform are of four types: daemons, clients, measurement plugins and third-party probes. The daemons are in charge of the orchestration of the measurements requested by the clients, by managing the plugins and interacting with the other daemons; the actual measurements are performed by means of the daemon-managed plugins cooperating with other plugins or third-party probes. UANM considers the plugins as gray-boxes, using only high level methods such as initializePlugin(), startMeasure(), ignoring underlying details. This eases the transformation of third-party tools in UANM-plugins with minimal changes to the original code and retains full compatibility of the UANM-plugin version and the original standalone version of the tool. Moreover, this allows UANM to be not tied to a single type of measurement or technique. Currently, we are focusing on available bandwidth measurement because we believe that this research area can particularly benefit from this platform. Therefore, the following plugins have been currently implemented: Abing [25], Assolo [14], Diettopp [20], IGI [17], pathchirp [29], Pathload [19], Spruce [34], Wbest [23]. The interactions between clients and daemons follow a client-server paradigm, while inter-daemons communications happen on peer-to-peer basis; both types of communication use a dedicated control protocol. An API in C allows external applications and platforms to act as clients. The communications between measurement plugins and with third-party probes use the plugin-specific control protocol. A diagram of the possible interactions among the components of the platform is reported in Fig.1. To illustrate how UANM works, in the following we describe the typical sequence of actions for a measurement experiment and the components involved.

- The client issues a measurement request to a daemon that runs on one of the edges of the path under test. The request can specify measurement constraints according to the intended purpose of the measure (e.g. server selection as described in Sec. 4). As an example, to perform server selection an application may request quick-and-dirty estimations towards the possible servers, while for continuous network monitoring a series of non intrusive measurements could be desired. Detailed constraints can be specified such as the averaging timescale, the number of subsequent estimations, the total probe load, or even the specific measurement technique and its parameters.

- The daemon performs a feasibility check to assess whether the request can be fulfilled and, in case the client did not request a specific one, which plugin is best fitted to the current *context*. The *context* comprises a description of the optional measurement constraints as well as the information about the measurement process, both structural (e.g. wireless hops or broadband access links along the path, UANM instances or known third-party estimation tools on the other edge, etc.) and behavioral (e.g. congested path, rapidly changing routes, etc.). This phase may imply communications with the other instrumented edge in order to update the *context*. If no constraints have been specified, the measurement will be set up to reach a trade-off between accuracy and intrusiveness. A daemon module called *decision engine* uses this information to select and configure the suitable plugin.
- A daemon module called *scheduler* schedules the measurement according to the policies (currently FCFS). This phase is needed because active estimation tools can be unreliable in case of concurrent measurements that involve shared resources (see Sec.3.2 for further details).
- The actual measurement process is performed executing the measurement phase of the plugin, in mutual exclusion with other possible measurements scheduled on the same path or on the same daemon.
- The result of the measurement is returned to the client, and it is also used to update the known *context*, allowing for better future decisions.

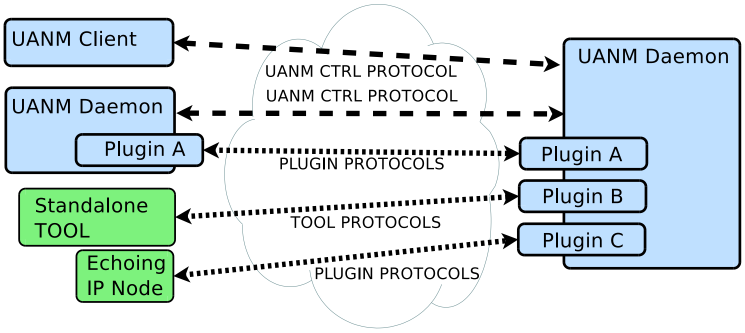


Fig. 1: UANM components and communications.

UANM is geared towards wide adoption among application developers and researchers in the field of network measurement. A prototype is released under the GPL terms, with a LGPL API for the development of measurement plugins. We believe that UANM allows to overcome the main limitations of current tools, providing accurate available bandwidth estimation in heterogeneous environments.

3.2 UANM highlights

UANM has been conceived with two main objectives: (i) the avoidance of the effect of the interference among concurrent measurement processes, and (ii) the automatic selection of a tool and its parameters in order to increase the performance. Moreover, offering a common generic model for the measurement tools (the *plugin API*) allows for a fair comparison of the different techniques in terms of probing time, intrusiveness, and accuracy in each given scenario. As for the first point, the problem a measure may encounter when more uncontrolled measurement processes share even one single part of the network for a long or short time interval has been shown by different works in literature [4–6, 2, 1]. This problem may occur since the current methods and tools do not provide coordination among measurement stations or any kind of alert feedback from the network.

In our former paper [1] we have shown the results of experiments aimed at assessing the capability of UANM in avoiding interference from concurrent measurements. In such experiments, using some of the most used available bandwidth estimation tools alone and concurrently, we experimentally verified and quantified the interference effect: in the concurrent case, the error is up to 3.5 times the error in the stand alone one. We also noticed that, in some experiments, the tools (Pathload in particular) did not converge to a final stage if run concurrently. This result has been attributed to the approach adopted by this tool, which leads the network towards congestion. Finally, we could assess that the design of UANM daemon avoids the interference effect thanks to the scheduler that coordinates the different clients. We refer the interested readers to [1] for more details.

The second aspect we highlight is the lack of measurement accuracy that may happen when the available bandwidth tools are used without knowledge of the basic network configuration. In fact, currently available tools are designed or tuned by default for a given scenario (layer 1 technology, intrusiveness of the measurement traffic, desired accuracy, maximum measurement time), and therefore are hardly suitable for as-is integration in third party software and automated use, as the technical knowledge of the tool and of the possibly varying scenario could be missing. For instance, pathchirp gives wrong result on high speed networks unless some of its parameters are correctly set up. The decision engine in UANM is able to automatically select and configure the best tool for this scenario. In pursuing the second objective the design of UANM follows the autonomic paradigm [18] as the interface presented to the user can be characterized as *Sensor* (providing the results of the measurements) and *Effector*, allowing the user to specify the *policies*, not the execution details of the measurements (though the latter possibility is offered as a special case for the experimenter).

The outcome of the request issued by the user is the result of a continuously running sequence that follows the autonomic control flow, presenting

- a *knowledge base* containing the current context (the available measurement tools and their characteristics, as well as the available information about the network)
- a *planning* phase (managing the scheduling of possible concurrent measurements)
- the *execution* of the scheduled plan (activation of the measurement tools)
- the *monitoring* of the results of the measurements and their *analysis* (that update the context and offer a data report to the user that requested it).

To evaluate experimentally the efficacy of the automatic management of the measurement tools, we compared the accuracy achievable with pathchirp or Pathload with that achievable when these tools are used through UANM on 1Gbps Ethernet links. The results evidenced that pathchirp and Pathload are quite inaccurate (with relative errors up to 90%) with the former being the least accurate. In this scenario UANM selected pathchirp with a different configuration of the packet trains, obtaining the highest accuracy in all the tested load conditions (with relative error 19% where the standalone configuration had 90%). We refer the interested readers to [1] for more details.

3.3 UANM: comparing available bandwidth estimation tools

In this section we present the results of a fair comparison of the performance of several available bandwidth estimation tools, performed by means of UANM.

Testbed and Tools These experiments have been conducted on the laboratory testbed depicted in Fig. 2. The end hosts are provided with *pathchirp* and *Pathload*, as well as with UANM (equipped with the eight plugins reported in Section 3.1). On the end hosts, we also installed a traffic generator called D-ITG [7] to generate the cross traffic in order to reproduce different network load conditions. Different hosts are used for cross- and probe-traffic generator in order to avoid interference between these activities.

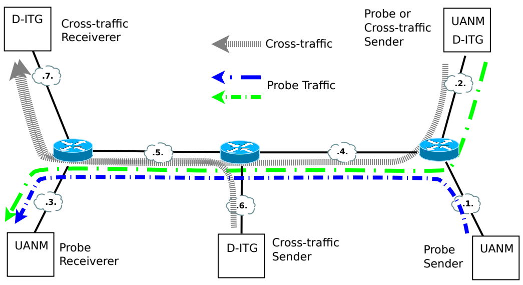


Fig. 2: Testbed used for the comparison of the performance of the tools.

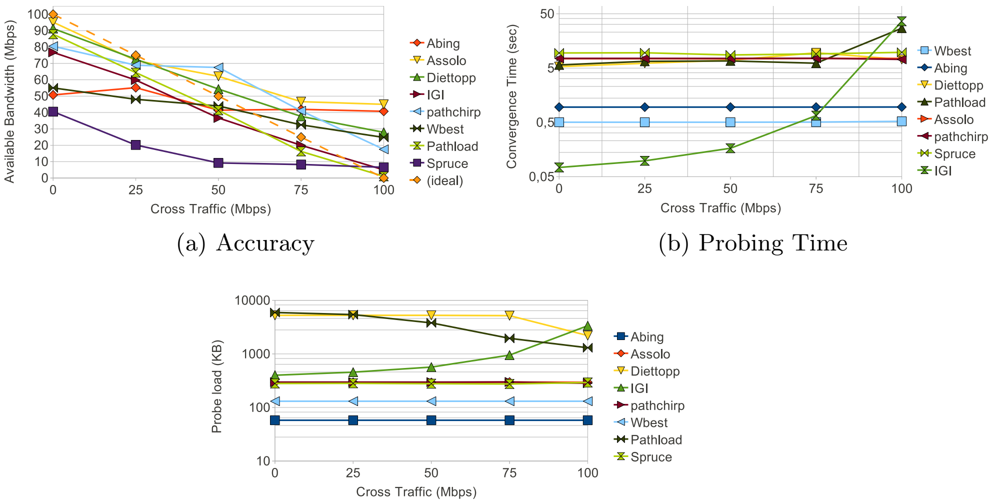
As we wanted to compare the techniques implemented by the tools in their standard operating conditions, we run all the plugins with default values of the configuration parameters. For the same reason, no specific settings have been enforced for the operating systems of the testbed nodes, and no change have been made on standard settings for the network adapters.

All the links in the testbed run at 100 Mbps, full duplex. CBR cross traffic has been generated by the host *Cross-traffic Sender* (bottom, center in the testbed) towards the host *Cross-traffic Receiver* (top, left in the testbed) at rates of 25, 50, 75 and 100 Mbps, and with IP packet size of 1000 Bytes. 10 measurements have been performed with each plugin and for each cross traffic rate, as well as with no cross traffic. The results shown in the following represent the average values of the 10 measures collected.

Results In Fig. 3a we report the results obtained by the tools in terms of accuracy. As we can see, Abing, Wbest and Spruce achieved the worst performance. A recent work in literature [15] reported slightly different results. A deeper investigation revealed that the difference is due to the fact the whole testbed is equipped with Gigabit Ethernet network adapters, whose interrupt mitigation feature is known to affect the performance of some available bandwidth estimation tools [32, 28]. Besides that, other important observations can be done: i) Wbest default parameters are suited to IEEE 802.11 wireless networks, therefore, its performance may be impacted by a full wired scenario; ii) Diettopp obtained the highest accuracy and also the smallest standard deviation; iii) Pathload also obtained accurate results, but the standard deviation is large, especially for small volumes of cross traffic; iv) Pathchirp achieved intermediate performance, both in terms of relative error and standard deviation; v) IGI/PTR obtained results similar to those of Pathload, and a small standard deviation.

Figure 3b shows the *probing time* of the tools, calculated as the difference between the timestamps of the last and the first probe packet: we consider this value as it represents both the time during which the network is solicited with additional traffic (for the calculation of intrusiveness), and the actual measurement interval (for the calculation of time averages and sampling rates); the time needed for the setup of the control channel and the mutual exclusion are not included in this value. In this figure, we can observe that the *probing time* of the tools is almost constant with the cross traffic rate, with the exceptions of IGI/PTR and Pathload. The probing time of IGI/PTR is increasing with the cross traffic: from 74ms with unloaded path, to 0.7s with 75 Mbps of cross traffic, and up to to 36 s with fully saturated path. Pathload obtained about the same probing times (i.e. 6 s) in almost all the load conditions, except at 100 Mbps, where it required about 27 s. The probing times of Assolo, Diettopp, Pathchirp and Spruce are comparable with those of Pathload ($\in [6, 8]$ s) with very low standard deviation.

Fig. 3c shows the volume of probe traffic generated by the tools for a measurement (in average). We notice that, besides IGI/PTR and Pathload, the volume is almost independent of the cross traffic rate. The intrusiveness of IGI/PTR in-



(a) Accuracy

(b) Probing Time

(c) Intrusiveness

Fig. 3: A fair comparison among eight available bandwidth estimation tools in terms of accuracy, probing time (between first and last probing packet) and intrusiveness (total *volume of probe traffic* injected). The standard deviation is not visible in the plots. Its figures are reported in the text for interesting cases.

creases with the cross traffic volume: from 410 KB to 3.4 MB, and the standard deviation increased from 46 KB to 980 KB. Pathload has a probe-traffic volume decreasing with the cross traffic rate: from 6 MB down to 1.3 MB, and the standard deviation decreased from 2.8 MB down to 208 KB. This behavior is due to the way the technique works. Having to congest the path, Pathload generates a larger volume of probe traffic when the network is unloaded. Pathload and Diettopp are the most intrusive tool, while Abing is the least intrusive one.

Thanks to this fair comparison, a few considerations can be done. The most accurate tools are Diettopp and Pathload, both being also the most intrusive and the slowest (i.e. highest probing time). IGI/PTR is the fastest in all the cases except with the fully saturated path, and it has a good accuracy. But, its probing time can range in more than 2 orders of magnitude. Pathchirp, while being less accurate for low cross-traffic volumes, has a probing time comparable with Pathload and Diettopp, but it is also less intrusive. Assolo has performance similar to Pathchirp, but higher accuracy for smaller volumes of cross traffic. Even if we used only the default settings of the plugins and a basic network scenario (varying only the cross traffic rate), the experimental results showed that there is no tool that suits the requirements of all the applications. Therefore, an informed choice of the tool and of its parameters is necessary for the effective and efficient estimation of the available bandwidth. UANM has been conceived with this idea in mind, and it can be profitably be used for this aim.

4 Application of Available Bandwidth Estimation

In the case of content distribution networks supporting IPTV, Video on Demand (VoD) or similar applications, there is a necessity to adhere to Quality of Service (QoS) targets. To this end, admission control is an important component to the deployment of a successful IPTV/VoD solutions. In order to show a possible application of the available bandwidth estimation, this section discusses how this parameter can be used in an admission control framework [21]. Such a framework operates in an end-to-end manner without the need to access measurements directly from the network topology connecting end-user points of attachment to those of one or more content servers. In the presented framework Quality of Service is maintained by performing admission control based on the estimated current available bandwidth of the network paths.

4.1 IPTV Admission Control using Available Bandwidth Estimates

The IPTV admission control framework is concerned with ensuring the adequate delivery of content from the content servers to the clients over an intermediate topology that is not controlled by the service provider (e.g. the Internet). Content servers have two purposes: to serve content items and to measure the available bandwidth between the content server and the destination edge router of the network. The edge router is also the point of attachment of the clients to the network. One server in the framework, referred to as the selection server, has the sole responsibility for collating the available bandwidth estimates for each path and making the decision on whether or not to accept a new request. This server works in conjunction with any number of content servers in the framework. For simplicity we assume that each content server contains a complete library of all the content items, allowing any item to be served from any server. An estimation of the available bandwidth between the the content server and the edge router is performed at regularly defined intervals. The results are continuously sent to the selection server. Independently of this, when a request is generated, it goes from the client to the selection server. The selection server makes a decision to accept or reject the request (based on the algorithms discussed in the following sections). A rejection is reported to the client, whereas an acceptance is delegated to the appropriate content server and the content is served.

A Video on Demand content library can be expected to contain in the order of hundreds or thousands of different content items. The algorithm presented here places each item into categories and operates by dealing with these categories. This allows the algorithm to remain efficient whilst still dealing with a large scale content library as the amount of categories would be expected to be in the order of tens for even a very large content library. Items can be categorized into groups with similar durations, bandwidth requirements, and revenue potential or a subset of these characteristics. Two content items that have a similar duration and peak throughput might be placed into different categories due to one being a newer release and therefore having a higher earning potential. Similarly, two items might be categorized differently despite having the same revenue potential if the durations are different enough to vary the cost involved in serving each content item.

We examine the performance of this framework under varying background traffic conditions within the network. We assume each content server has the resources required to serve any content assigned to it and that all flows assigned to a server run to completion. This allows us to focus on the performance of the admission control framework and not the processing capabilities of the content servers.

4.2 Available Bandwidth based Server Selection / Admission Control Algorithm

A simple selection/admission control algorithm bases its decision to accept or reject a new request for a content item type on whether any of the content servers have the available bandwidth required on their path to the client. Assume there are I individual types of content made available by the service provider. Let $i = 1, \ldots, I$ denote an arbitrary type of content item. Let $p(i)$ denote the peak bandwidth per second required by item type i. Assume that the service provider maintains J content servers, each with a single dedicated egress link to the core network. Let $j = 1, \ldots, J$ denote an arbitrary content server. Let $\hat{B}_{jd}(t)$ denote the estimate of available bandwidth between content server j and edge router d as calculated at time t.

As mentioned previously, the selection server maintains an estimate of the Available Bandwidth $\hat{B}_{jd}(t)$ for the current time t of each path between content server j and edge router d. This estimate is calculated as a moving average of recent reported estimates. If only one server is listed as possessing enough bandwidth to support a request for a particular item type, then the request is accepted and allocated to that content server j^*. If there are multiple servers capable of supporting the request, j^* is assigned to be the content server with the highest available bandwidth. The final case occurs when none of the network paths have sufficient bandwidth and in this case the request is rejected. Once accepted to a server, a traffic flow will use this server for the duration of the flow. This is specified formally in Alg. 1.

4.3 Simulations and Results

The framework has been firstly implemented in simulation to carefully test its performance and evaluate the impact of different parameters before the deployment in real scenarios. The simulations were performed using the OPNET Modeler$^{\text{TM}}$[26] simulation environment. The framework is deployed in a scenario where there is three different content servers (A,B,C). We use traffic traces taken from actual videos using various CODECs by [30] and use [11] to inform the distribution of mean durations, enabling us to create realistic traffic flows. To ensure a conservative use of available bandwidth at each server, a threshold of 90% is used at each server as an upper bound. This is to cater for multiple flows reaching peak throughput simultaneously. For the purposes of control and analysis,

```
Input: $i^*, \{\hat{B}_{jd}(t)\}$

forall content servers $j = 1 \dots J$ do
    List all content servers $\{j'\}$ for which $\hat{B}_{jd}(t) > p(i^*)$;
    if $\{j'\} \neq NULL$ then
        Select $j^* \in \{j'\} : \hat{B}_{j^*d} = \max\{\hat{B}_{j'd}(t)\}$;
        return ACCEPT, $j^*$;
    end
    else
        return REJECT;
    end
end
```

Algorithm 1: Available bandwidth admission control algorithm (ABAC)

we concentrate on requests arriving into the network from a single access point and we specify the intermediate topology between the content servers and the access point to contain Fast Ethernet (100Mbps) links.

For the estimation of the Available Bandwidth in this scenario we have to consider the specific (often contrasting) requirements. Firstly, higher probing rates lead to more accurate results, but the result takes longer to be generated and leave a heavier footprint on the network as the tool will inject more data into the network. Secondly, the available bandwidth estimation frequency, or, conversely, the inter-estimate time, can influence how long it takes for the estimate to update after a change in the bandwidth that is available: the shorter the time between estimates, the faster a tool can become aware of changes in the available bandwidth, but also the more intrusive is the probing. The application implementing the Server Selection / Admission Control algorithm would request UANM for a measurement profile characterized by relaxed constraints for probing time and accuracy, low intrusiveness and medium repetition frequency (once per video request for each server). The tool chosen by the UANM platform in the case of a wired path with capacity upper bounded by 100Mbps for this measurement profile would be Pathchirp [29] as indicated in Table 2 - *predefined measurement profiles* of [1], specifically the entry MONITORING. To confirm these findings, we also performed a simulation study with the tools available as plugins in UANM. The following consideration have been drawn from this study. First, increasing the spread factor (i.e. decreasing the probing rate) does not affect the QoS linearly. The number of flows accepted increases with a larger spread factor. However, the benefits of this are nullified by the significant increase experienced in end-to-end delay. Second, as the inter-estimate time lowers so does the end-to-end delay, an inter estimate time of one second requires a high overhead of control and probe packets. Therefore, in the rest of these simulations, aimed at showing the benefits of available bandwidth estimation for server selection, we utilize Pathchirp with spreading factor equal to 1.2 and inter-estimate time equal to 5 s. We refer the interested readers to [1] and [21] for further details.

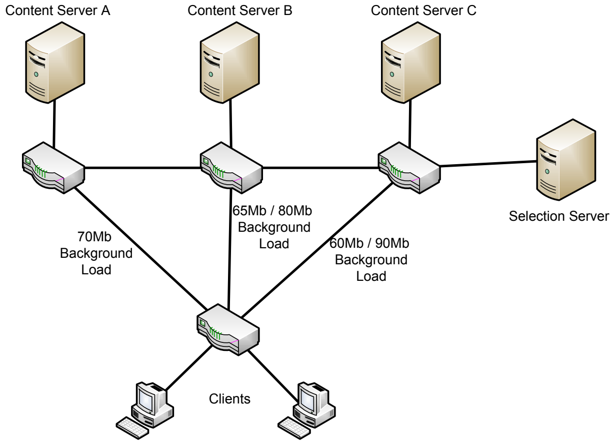
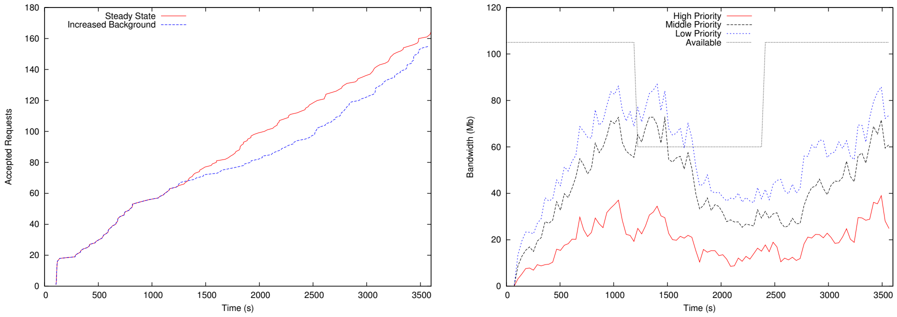


Fig. 4: Topology Used In Simulation Environment.

The following tests were carried out on the simulated topology, we analyze the performance of the algorithm when operating in both a steady state network environment with different kinds of background traffic and also in the situation where there is a degradation in the condition of the network, such as, for example, when the background traffic increases. The number of links between the content servers and the clients access point was kept equal at 2 hops and background loads were added to the paths by using OPNETs traffic generators. These traffic generators provided a base background traffic of 70Mbps, 65Mbps, and 60Mbps on the paths of the three content servers mentioned, as shown in Figure 4. The packet sizes for the background loads were uniformly distributed with an average of 576 bytes (IPv4 MTU). To create the increase in background traffic we introduce a step change to two of the traffic generators. The 65Mbps background load is increased to 80Mbps and the 60Mbps background load is increased to 90Mbps. This is also shown in Figure 4. Overall, this reduces the available bandwidth from 105Mbps down to 60Mbps with a significant change in where the majority of that bandwidth is available.

We compare the number of requests admitted to the network when available bandwidth estimates are used to analyze the connection quality against no such information is available. The results in Figure 5a depict that the use of available bandwidth estimates allows for a dynamic response to the changing bandwidth conditions within the network by lowering the number of requests admitted as the background traffic increases.

We also introduced a step function to simulate a dramatic change in background traffic to analyze how the admission control algorithm responded. The test consisted of three classes of video requests arriving different request rates. In Figure 5b all the classes of video see a reduction in the throughput they are generating. These simulations allowed to show how, relying on the available bandwidth estimation, this approach does not need access to the intermediate



(a) Requests admitted by ABAC.

(b) ABAC Utilization.

Fig. 5: ABAC Analysis

topology and the video traffic is robust to external changes in the available bandwidth as it adapts the number of requests admitted.

5 Conclusions

In this paper we have presented a very important metric for analyzing the quality of communication links, namely available bandwidth. We described the novel UANM platform that has been conceived for properly choose, configure, and use the large number of available bandwidth estimation tools available in literature. As available tools and techniques require tailoring to specific scenarios and require an expert operator to obtain accurate results, the platform facilitates detailed analysis of each tool under common network scenarios and can be used to identify appropriate configurations of tools to suit particular applications. Thanks to UANM, we have also shown the results of a comparison of the performance, accuracy, and intrusiveness of the mainly used tools for available bandwidth estimation, evidencing their pros and cons in different scenarios. Finally, we presented a server selection and admission control framework which bases decisions on measurements of available bandwidth between the client host and a server host. We chose to use Pathchirp with an appropriate configuration to suit our requirements, based on the analysis carried out through UANM. The framework demonstrated that using appropriately configured available bandwidth estimations can ensure that effective decisions are taken for both choosing an appropriate server to deliver video content to the client host, and also whether there is enough bandwidth available on the end-to-end path to host the video stream, without incurring any additional congestion within the network.

References

1. Aceto, G., Botta, A., Pescapé, A., D'Arienzo, M.: Unified architecture for network measurement: The case of available bandwidth. Journal of Network and Computer

Applications (2011)

2. Aceto, G., Botta, A., Pescapé, A., D'Arienzo, M.: UANM: a platform for experimenting with available bandwidth estimation tools. Computers and Communications, IEEE Symposium on pp. 174–179 (2010)
3. Ali, A.A., Michaut, F., Lepage, F.: End-to-End Available Bandwidth Measurement Tools : A Comparative Evaluation of Performances. In: 4th Intl Workshop on Internet Performance, Simulation, Monitoring and Measurement, Austria, 2006 (Jun 2006), http://arxiv.org/abs/0706.4004
4. Angrisani, L., Botta, A., Pescapé, A., Vadursi, M.: Measuring wireless links capacity. In: Wireless Pervasive Computing, 2006 1st International Symposium on. pp. 5 pp.+ (2006), http://dx.doi.org/10.1109/ISWPC.2006.1613635
5. Botta, A., D'Antonio, S., Pescapé, A., Ventre, G.: BET: a hybrid bandwidth estimation tool. In: Proceedings of the 11th International Conference on Parallel and Distributed Systems, 2005. vol. 2, pp. 520–524 (2005), http://dx.doi.org/10.1109/ICPADS.2005.103
6. Botta, A., Pescape, A., Ventre, G.: On the performance of bandwidth estimation tools. In: Systems Communications, 2005. Proceedings. pp. 287–292. IEEE (2005)
7. Botta, A., Dainotti, A., Pescapé, A.: A tool for the generation of realistic network workload for emerging networking scenarios. Computer Networks 56(15), 3531 – 3547 (2012)
8. Botta, A., Pescapé, A., Ventre, G.: An approach to the identification of network elements composing heterogeneous end-to-end paths. Computer Networks 52(15), 2975 – 2987 (2008)
9. Botta, A., Pescapé, A., Ventre, G.: Quality of service statistics over heterogeneous networks: Analysis and applications. European Journal of Operational Research 191(3), 1075 – 1088 (2008)
10. Cabellos-Aparicio, A., Garcia, F., Domingo-Pascual, J.: A novel available bandwidth estimation and tracking algorithm. Proceedings of IEEE Network Operations and Management Symposium pp. 87–94 (2008)
11. Cheng, X., Dale, C., Liu, J.: Understanding the characteristics of internet short video sharing: Youtube as a case study. ACM SIGCOMM Conference on Internet Measurement p. 28 (2007)
12. Chun, B., Culler, D., Roscoe, T., Bavier, A., Peterson, L., Wawrzoniak, M., Bowman, M.: Planetlab: an overlay testbed for broad-coverage services. ACM SIGCOMM Computer Communication Review 33(3), 3–12 (2003)
13. Geni-global environment for network innovations. http://www.geni.net (Sep 2012)
14. Goldoni, E., Rossi, G., Torelli, A.: Assolo, a new method for available bandwidth estimation. International Conference on Internet Monitoring and Protection pp. 130–136 (2009)
15. Goldoni, E., Schivi, M.: End-to-end available bandwidth estimation tools, an experimental comparison. In: Proceedings of TMA'10, 2010. pp. 171–182 (2010)
16. Guerrero, C.D., Labrador, M.A.: Traceband: A fast, low overhead and accurate tool for available bandwidth estimation and monitoring. Computer Networks 54(6), 977–990 (April 2010), http://dx.doi.org/10.1016/j.comnet.2009.09.024
17. Hu, N., Steenkiste, P.: Evaluation and characterization of available bandwidth probing techniques. IEEE Journal on Selected Areas in Communications 21, 879–894 (2003)
18. IBM Corporation: An architectural blueprint for autonomic computing. Autonomic Computing, white paper (2006)

19. Jain, M., Dovrolis, C.: End-to-end available bandwidth: measurement methodology, dynamics, and relation with tcp throughput. IEEE/ACM Trans. Netw. 11(4), 537–549 (August 2003)
20. Johnsson A., Melander B., B.M.: Diettopp: A first implementation and evaluation of a simplified bandwidth measurement method. In: Proceedings of the 2nd Swedish National Computer Networking Workshop (2004) (2004)
21. Meskill, B., Davy, A., Jennings, B.: Server selection and admission control for IP-based video on demand using available bandwidth estimation. IEEE Conference on Local Computer Networks (2011)
22. Meskill, B., Davy, A., Jennings, B.: Revenue-maximizing server selection and admission control for iptv content servers using available bandwidth estimates. in Proc. IEEE/IFIP Network Operations and Management Symposium (2012)
23. Mingzhe, L., Claypool, M., Kinicki, R.: Wbest: A bandwidth estimation tool for ieee 802.11 wireless networks. 33rd IEEE Conference on Local Computer Networks (LCN) pp. 374–381 (October 2008)
24. Murray, M., Smallen, S., Khalili, O., Swany, M.: Comparison of end-to-end bandwidth measurement tools on the 10GigE TeraGrid backbone. In: The 6th IEEE/ACM International Workshop on Grid Computing, 2005. pp. 300–303 (2005)
25. Navratil, J., Cottrell, R.L.: Abwe: A practical approach to available bandwidth estimation. In: in Passive and Active Measurement (PAM) Workshop 2003 Proceedings, La Jolla (2003)
26. OPNET: Discrete event smulation model library. OPNET Modeler^{TM}(2011), http://www.opnet.com/
27. Paxson, V.: Strategies for sound internet measurement. In: Proceedings of the 4th ACM SIGCOMM conference on Internet measurement. pp. 263–271. IMC '04, ACM, New York, NY, USA (2004), http://doi.acm.org/10.1145/1028788.1028824
28. Prasad, R., Jain, M., Dovrolis, C.: Effects of interrupt coalescence on network measurements. In: In Proceedings of PAM '04 (2004)
29. Ribeiro, V., Riedi, R., Baraniuk, R., Navratil, J., Cot, L.: pathchirp: Efficient available bandwidth estimation for network paths. In: Passive and Active Measurement Workshop, 2003 (2003)
30. Seeling, P., Reisslein, M., Kulapala, B.: Network performance evaluation using frame size and quality traces of single-layer and two-layer video: A tutorial. IEEE Communications Surveys Tutorials 6(3), 58 –78 (2004)
31. Shriram, A., Murray, M., Hyun, Y., Brownlee, N., Broido, A., Fomenkov, M., Claffy, K.: Comparison of public end-to-end bandwidth estimation tools on high-speed links. In: Proceedings of PAM 2005. vol. 3431, pp. 306–320 (2005)
32. Sommers, J., Barford, P., Willinger, W.: A proposed framework for calibration of available bandwidth estimation tools. In: Computers and Communications, 2006. ISCC '06. Proceedings. 11th IEEE Symposium on. pp. 709 – 718 (june 2006)
33. Song, H., Zhang, Q.: Netquest: A flexible framework for large scale network measurements. IEEE/ACM Transactions on Networking 17(1), 106–119 (2007)
34. Strauss, J., Katabi, D., Kaashoek, F.: A measurement study of available bandwidth estimation tools. ACM SIGCOMM IMC pp. 39–44 (2003)
35. Xu, D., Qian, D.: A bandwidth adaptive method for estimating end-to-end available bandwidth. In: 11th IEEE International Conference on Communication Systems. pp. 543–548 (2008)