

2005

An Efficient Accounting Architecture for QoS-aware Internet Traffic

Abdelnasser M. Abdelaal

University of Nebraska at Omaha, abdelnasser.ibr@cas.edu.com

Hesham H. Ali

University of Nebraska at Omaha, hali@mail.unomaha.edu

Follow this and additional works at: <http://aisel.aisnet.org/amcis2005>

Recommended Citation

Abdelaal, Abdelnasser M. and Ali, Hesham H., "An Efficient Accounting Architecture for QoS-aware Internet Traffic" (2005). *AMCIS 2005 Proceedings*. 23.

<http://aisel.aisnet.org/amcis2005/23>

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 2005 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

An Efficient Accounting Architecture for QoS-aware Internet Traffic

Abdelnasser M. Abdelaal

Department of Computer Science
College of Information Science and Technology
University of Nebraska at Omaha
Omaha, NE 68182
aabdelaal@mail.unomaha.edu

Hesham H. Ali

Department of Computer Science
College of Information Science and Technology
University of Nebraska at Omaha
Omaha, NE 68182
hali@mail.unomaha.edu

ABSTRACT

The widespread usage of Internet-based applications and the large diversity of these applications, particularly in terms of Quality of Service (QoS) requirements, necessitate developing a QoS-aware Internet accounting architecture. Developing such a architecture has grown to be increasingly essential for service pricing, cost recovery, resource management, and network analysis. This paper proposes an efficient Internet accounting architecture that employs various packet, flow, link, and end-to-end QoS parameters in a way that allows for a simple and practical implementation. With the help of the proposed architecture, a number of eBusiness applications and advanced network-tracking techniques can be efficiently developed.

Keywords

Internet accounting, business models, service pricing, network management, IntServ

INTRODUCTION

Pervasive computing and QoS-based eServices via the Internet have become ever more essential in helping to accomplish daily life affairs. Accounting for phone subscribers is much easier because the phone numbers include information about the subscribers' locations and identities. However, IP addresses do not indicate such information which makes accounting for eServices not just essential, but also so sophisticated and costly. Managing and controlling the Internet traffic is subject to meeting the requirements of Internet Service Providers (ISPs) and the expectations of users based on service differentiation. This service differentiation requires measuring and reporting users' usages of the Internet resources to use this collected information for business and management purposes. Recently, the attention of per-flow (per-connection) treatment and end-to-end QoS guarantee has grown for several reasons:

1. The emergence of content-based applications
2. Increasing VoIP applications
3. Diversity of Internet applications in terms of their QoS requirements
4. The features of the foreseen 4G Internet

Thus, an efficient end-to-end QoS-aware Internet accounting architecture is crucial to meter and record information about the subscribers' attributes and the network usages. This information includes the user's identity, the connection start and end time, traffic volume (transmitted and received), class of service, connection duration, time of day, source, destination, and type of application [1]. The recorded parameters provide ISPs with information about subscribers, resources, and system conditions. These parameters facilitate service pricing, cost recovery, expansion planning, traffic monitoring, reporting, auditing, traffic analysis, and intra-domain settlements [18]. The difference between bandwidth metering and traffic accounting is that traffic accounting relates packets and applications to the users' behaviors in a way that interacts with the system conditions. The current bandwidth and resource metering tools, mostly packet-based, are limited to three parameters: available bandwidth, link capacity, and bulk transfer [7]. In addition to providing users with information about their Internet usage attributes, Internet accounting enables users to interact with the system in a way that maximizes their benefits and improves the system performance. However, traffic metering and accounting are expensive in terms of the operation and management overhead.

The exponential increase of internet applications and their diversity, in terms of their QoS requirements, have evolved many accounting applications and eBusiness models such as dynamic Internet pricing [1], content-based billing [9], reverse-charging [19], and roaming and intra-domain settlements [18]. Such models augment the awareness of per-flow, or per-

request, end-to-end treatment. We have chosen IntServ QoS architecture as a platform for our proposed architecture because it is superior to DiffServ in terms of its per-flow treatment and end-to-end QoS guarantee [16, 3].

PREVIOUS WORK

The Internet Engineering Task Force (IETF) has few working groups that are working on developing Internet traffic accounting architectures. For example, The Real-Time Traffic Flow Measurement (RTFM) working group has developed a flow-aware accounting architecture that counts the total number of packets for each flow and determines the source and the destination for each packet [4]. The Cisco Netflow accounting server collects the following parameters about each flow: the IP addresses for the source and the destination, flow duration, number of packets and bytes for each flow, used protocol, class of service, and the switch port [10]. However, the Netflow server does not support real-time routing and its accounting process is based on randomized samples of the system traffic. In addition, it only tracks the class of service for each packet based on a per-hop behavior (PHB), not end-to-end. Developing eBusiness applications such as QoS-aware billing based on a PHB or randomized samples is not reliable.

The Authorization, authentication, and accounting (AAA) architecture collects and reports information such as users' identities, executed commands, number of bytes, number of packets, consumed resources, and connection start and end time [2]. However, we believe that calculating the number of packets and the number of bytes for each flow is redundant. AAA sends this information to the server for security, reporting, billing, and auditing purposes. There are some other authentication, authorization and accounting technologies such as DIAMETER, RADIUS, and ATM server. For a literature review of the current Internet accounting technologies, we refer the reader to [11, 14, and 15]. Based on the literature review, we found out that the current accounting architectures do not collect information about end-to-end QoS parameters such as packet drop rate, packet delay, reserved resources for each flow, and end-to-end path characteristics.

Internet billing is one of the most important eBusiness applications for Internet accounting [13]. Edell et al. have developed the pilot billing server for TCP traffic [8]. The system collects data on real-time usage over T-1 lines at Berkeley University. The main idea is to trace the setup and tear-down of TCP sessions which are supported by users' authentications to charge them based on the connection duration. The NZGate (New Zealand Gate Way) billing server identifies the incoming flows and outgoing flows and charges them based on their volume [5]. NZGate adapts the concept of committed traffic volume based on which subscribers are charged. The traffic volume is measured, monitored and reported every month for each user. It offers a 30% discount for low priority traffic (e.g. emails, FTP) and an 80% discount for off-peak usages. In addition, NZGate informs subscribers about the cost and the benefits of different usage options so that they could budget for their usages. The Monash University billing and accounting server [20] does not charge students and faculty who use the server's mailing system, but it charges them if they use any external mailing system such as Hotmail or Yahoo.

A NEW APPROACH FOR INTERNET ACCOUNTING

We propose an efficient end-to-end QoS-aware Internet accounting architecture that can fit the next generation of Internet applications. The contribution of this paper is twofold: the first is presenting a framework for designing an efficient Internet accounting architecture; the second is embedding this efficiency framework in a new Internet accounting architecture for per-flow end-to-end QoS guarantee.

The proposed efficiency properties

An efficient Internet accounting architecture should have the following properties:

1. Optimal grain size
2. End-to-end tracing
3. Fairness
4. Completeness
5. No redundancy

Accounting aggregation, or the grain size of data recording, is very important especially if multiple ISPs/operators are involved. The granularity of the accounting architecture decides whether data recording will be based on per-packet, per-session, or per-link level. Therefore, an efficient accounting architecture should only collect information about useful parameters and should avoid *redundancy* to avoid excessive overhead. Therefore the *optimal grain size* is very important to minimize the operational and management overhead.

NSFNET transmits 60 billion packets every month on average, which in turn, requires 60 billion data records. To reduce the accounting management cost, NSFNET collects information from only a sample of the packets from sub-networks connected

to the backbone. In addition, it is expected that the routers of the next generation Internet will carry Terabits per second that will make packet-based accounting even more expensive. Therefore, the granularity of the Internet accounting policy is very important [14]. The operation cost of the accounting policy is the essential factor to determine how practical and efficient it is for eBusiness applications. These costs include reporting costs, collecting overhead, processing overhead, and security overhead. The optimal granularity level of the accounting policy should balance between the costs and the benefits of adapting this accounting policy.

In best-effort and DiffServ networks, which are PHB, the QoS varies in terms of packet delay, allocated resources, and the class of service from hop to hop. DiffServ uses Traffic shaping to overcome this problem [22]. Therefore, an *end-to-end* accounting architecture is crucial if the system involves many ISPs and seamless network clouds to report the QoS variation among these different clouds. In addition, the *end-to-end* property is important in order to determine which network domain is going to collect the accounting information, to help determine where it should be stored, and to explain how ISPs can exchange this information to use it for different purposes. Such property is essential for the 4G internet whose main feature is end-to-end QoS support for IP traffic in seamless environments [21]. We assume that the end-to-end packet delay is bounded to 1 ms, and we have used the Resource Reservation Protocol (RSVP) for end-to-end resource reservation. The end-to-end property identifies inter-domain and intra-domain traffic to treat them differently with respect to security, billing, and QoS guarantee. It is apparent that traffic remaining within the same domain should have soft QoS guarantee and should be charged less or even be free of charge. On the other hand, the outgoing traffic should be more secure and have a hard QoS guarantee. Moreover, these end-to-end accounting parameters facilitate intra-domain settlements among multiple ISPs. Moreover, the accounting architecture should be scalable to cover seamless edges without exponential increase of the accounting costs.

The importance of accounting *fairness* arises in the following cases: different QoS requirements; the need for reverse charging; the need for content-based billing; and the interaction between the user and the system conditions. In a reverse charging system, the accounting architecture should record the user who sends the RSVP message so he/she can be charged for this connection. For example, if a student is holding a video conference with a professor, the student is responsible to pay for this connection. In addition, packet-based charging makes subscribers pay for bad routing and switching technologies used by ISPs. In addition, retransmitted packets increase the overhead and the operation costs of accounting. In such a case, we believe that flow-based pricing is fair enough comparing to packet-based pricing.

The *completeness* of an accounting architecture should not be only in terms of the collected parameters but also in terms of the relationship among these parameters. Specifically, it should identify the relationship among link parameters, packet parameters, and per-request parameters. For example, the proposed architecture calculates the connection success rate, the connection failure rate, and the packet drop rate related to the link utilization level. However, there is a tradeoff between *completeness* and *redundancy*. For instance, Cisco NetFlow and AAA servers calculate the number of packets and the number of bytes for each flow. In spite of its *completeness*, this process has *redundancy*. Another example is the tradeoff between security and QoS guarantee. If the whole packet is encrypted, including the header, there is no way to determine and record its class of service. The accounting architecture should be dynamic and selective concerning the collected parameters for each user, service, and network domain.

The proposed architecture

The proposed accounting architecture encapsulates the previously discussed efficiency requirements so it can fit the foreseen 4G Internet applications. This architecture combines link accounting, per-flow accounting, and packet accounting in a way that is suitable for practical implementation of several eBusiness models and applications. For the purpose of simplicity, this architecture collects information about a few flow parameters (statistics) such as flow ID, start time, end time, flow duration, and flow connectivity. We assume that all flows require the same amount of resources. In addition, it collects information about packet parameters such as packet drop rate (PDR) and packet delay. Moreover, it records all the previous parameters with respect to link utilization level and it collects data about link status and the relationship between link utilization target and the actual achieved link utilization. The proposed architecture emphasizes the importance of the interaction among link accounting, packet accounting, and flow-based accounting in developing eBusiness applications and network management. Therefore, the proposed architecture is distinguished of its end-to-end QoS-awareness and it minimizes the management overhead for Internet accounting.

IMPLEMENTATION AND RESULTS

To evaluate the validity of the proposed architecture, we have used the IntServ and the Measured Sum (MS) [12] call-admission control module in NS-2.26 simulator [17]. IntServ classifies traffic into controlled-load service, guaranteed

service, and best-effort service. It provides end-to-end per-flow QoS, and it is supported by the RSVP. These properties make it suitable for multimedia applications (including VoIP) which require bandwidth on demand. We have used a simple topology of a single source and a single destination connected by a duplex-link of 10Mbps bandwidth. Packet transmission has a propagation delay of 1ms and size of 125 bytes. Flows have exponential arrival rates with an average of 400ms. The flow lifetime also has an exponential distribution with an average of 300sec. The used traffic generator model is an exponential on/off source with a peak rate of 64k. We have designed three experiments to measure link accounting, per-flow accounting, and packet accounting.

Link utilization level and accounting parameters

IntServ is a controlled-load architecture, where the system administrator can control the link load to achieve specific requirements. The simulation results, which we extracted from the trace file, in Table 1 shows that the system administrator can change the link utilization target (U.T.) to control the link utilization level (U'), connection admission rate (C.A.), connection rejection rate (C.R.), or the packet drop rate (PDR). In addition, it calculates the average flow lifetime (L.T.).

U.T.	U'	C.A.	C.R.	L. T.	PDR
98	91	5181	2466	164	8.2
95	88	5265	2383	158	4.5
92	85	5319	2329	152	0
90	84	5385	2263	148	0
88	83	5423	2225	144	0
83	78	5503	2145	142	0
85	75	5574	2094	138	0
80	74	5636	2012	133	0
78	73	5666	1982	129	0
70	65	5891	1757	114	0

Table 1 Link utilization and related accounting parameters

Table 1 shows the relationship among different accounting parameters and link utilization. It is obvious that when link utilization exceeds 92%, the network is highly congested and the PDR increases and the connection rejection rate (C.R.) increases as well. The actual link utilization (U') level is less than the utilization target (U.T.) of 6 points in average. Figure 1 shows the relationship between connection admission and rejection from one side and the link utilization from the other side. This relationship helps in expansion plans and congestion control.

Per-flow treatment and accounting

Per-flow, or per-request, treatment requires the flow ID, start and end time for each flow, the flow lifetime, the provided QoS parameters, and the reserved resources for each flow. In addition, it requires information about whether the flow has been admitted or rejected. Table 1 shows the average call admission rate and the call rejection rate related to the link utilization.

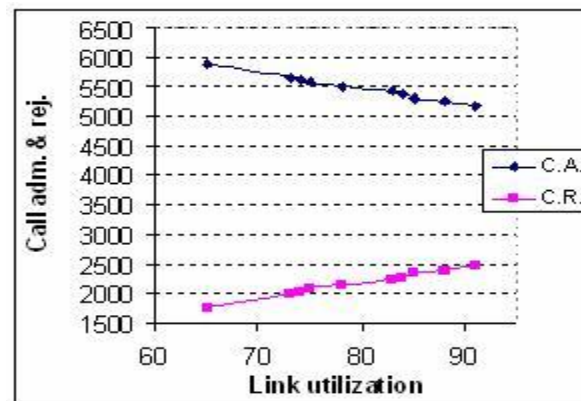


Figure 1 Call admission and rejection related to link utilization

QoS parameters are packet delay, which is bounded to 1ms in all experiments, and packet drop rate (PDR) which is shown in Table 1. The RSVP is used to reserve and guarantee resources for each flow, and we assume that all flows require the same amount of resources. We assume that each request (connection) is only one flow. These flow parameters could be called service statistics per request. This usage-data could be stored in XML files to be used for different applications. Table 2 shows the flow ID, start time, end time, and flow duration for each flow. Figure 2 shows the variation of connection duration for the first 10 connections. For example, flow number 4 occupies the resources for 7.2 sec. and flow number 10 occupies the resources for 285 sec. It is obvious that these flows should be treated differently in terms of price, priority, and security based on their duration.

Flow ID	Start time	End time	L.T.
1	1.072	867.806	866.7
2	1.318	639.533	638.2
3	1.575	259.835	258.3
4	1.722	8.886	7.2
5	1.969	30.109	28.1
6	2.051	269.822	267.8
7	2.146	44.036	41.9
8	2.157	51.217	49.1
9	2.876	56.768	53.9
10	2.884	285.030	282.2

Table 2 Per-flow statistics

Packet statistics and accounting

In general, Internet traffic could be either data applications or real-time applications. Data applications are sensitive to packet drop rate (PDR). Real-time applications are sensitive to packet delay and jitter. In this experiment, packet accounting traces only packet drop rate PDR and packet delay. The packet delay is bounded to 1ms which has been given as an input to the simulator. PDR from different simulation scenarios is illustrated in Table 1. Both packet parameters are called end-to-end packet statistics. To avoid redundancy, counting the number of packets for each flow has been ignored; because it has been assumed that all flows acquire the same amount of resources.

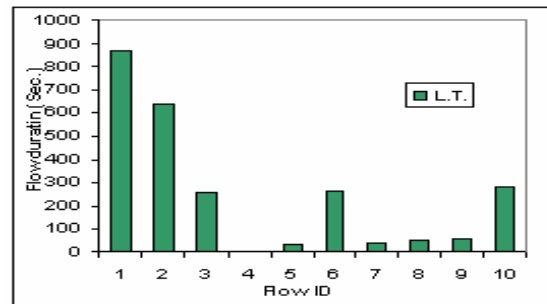


Figure 2 Connection lifetime for the first 10 flows

eBUSINESS MODELS AND APPLICATIONS

The proposed accounting parameters and the relationship among these parameters evolve several eBusiness and networking applications. For instance, end-to-end link statistics, such as the total number of flows and the PDR related to the link utilization helps in expansion and investment plans. Using statistics about congestion periods can classify traffic into two categories: peak traffic and off-peak traffic. Peak traffic could be charged more and off-peak traffic could be given a discount. In addition, ISPs could offer incentives to subscribers who can shift their bulk transfer to off-peak hours. The connection which gets disconnected should receive a delayed discount. Moreover, these accounting parameters could be used for pricing and billing policies. Using a controlled-load network, e.g. IntServ, is crucial to enforce service access for prepaid services and better facilitates different rating plans for different applications, e.g. music, games, images, TV trailers, etc, based on their content and their QoS requirements. End-to-end accounting information will help in minimizing the

accounting transaction costs among different ISPs. Traffic metering and accounting is important for efficient resource allocation and cost recovery and to enforce specific policies.

The per-flow parameters could be used for content-based billing and reverse charging. In addition, the ISPs could penalize inefficient resource usages and bad network configurations and reward efficient usages based on the per-flow analysis. Implementing this accounting policy over end-to-end architecture will allow ISPs for better intra-domain settlements, distributed resource allocation, revenue sharing, and dispute resolution. This flow-based information and accounting could be used for flow-based pricing, congestion control, and it also enables ISPs to target misbehaving and unresponsive flows and fast link-failure recovery. In addition, recording the start and end time stamp for a service is very crucial in web services [6] and transactions such as e-Auctions and stock quote services.

CONCLUSIONS AND FUTURE DIRECTIONS

In this paper, we have proposed a framework for designing an efficient QoS-aware Internet accounting architecture. The efficiency has been measured by optimal grain size, end-to-end tracing, fairness, completeness, and non-redundancy. Based on these properties we have designed an Internet accounting architecture which integrates per-packet, per-flow, and per-link accounting with respect to end-to-end QoS requirements and path characteristics. In addition, we illustrated the relationship among these accounting parameters to reflect a general view of the system performance. The obtained results indicate that this accounting architecture has low operational overhead. The proposed architecture is particularly appropriate for several eBusiness applications and network management such as billing, service differentiation, and congestion control.

The proposed architecture is mainly suitable for IP networks with controlled load. Our future research will focus on optimizing the accounting measuring and reporting mechanisms. Such mechanisms are particularly important in the wireless domain due to the limited bandwidth, high error rate, and high mobility and roaming. The scalability of accounting architectures and reporting additional QoS parameters such as packet jitter and service security will remain an area which requires further studies.

ACKNOWLEDGMENT

This research project was partially supported by NSF grant number EPS-0346476 and Nebraska Research Initiative (NRI) grant.

REFERENCES

1. Abdelaal, M. A. Ali, H. and Omran, H. (2004) A New Dynamic Pricing Policy for Integrated Service Networks, *The 1st International Computer Engineering Conference New Technologies for the Information Society*, December 27-30, Cairo, Egypt, Cairo University
2. Authentication, Authorization and Accounting (aaa) (2005). Retrieved February 18, 2005, from <http://www.ietf.org/html.charters/aaa-charter.html>
3. Blake, S. Black, D. Carlson, M. Davies, E. Wang, Z. and Weiss, W. (1998) An Architecture for Differentiated Services, RFC 2475, December, Retrieved February 10, 2005, from <http://www.ietf.org/html.charters/diffserv-charter.html>
4. Brownlee, N. Mills, C. and Ruth G., (1997) Traffic Flow Measurement: Architecture, RFC 2063, January 1997. Retrieved February 15, from <http://www.faqs.org/rfcs/rfc2722.html>
5. Brownlee, N. New Zealand Experiences with Network Traffic Charging. Retrieved January 20, from <http://www.press.umich.edu/jep/works/BrownNewZe.html>
6. Curbera, F. Khalaf, R. Mukhi, N. Tai, S. and Weerawarana, S. (2003) The next step in web services, *Communications of the ACM*, 46,10, 29-34.
7. Dovrolis, C. Prasad, R. Murray, M. and claffy, k. (2003) Bandwidth estimation: metrics, measurement techniques, and tools, *IEEE Networks*, November.
8. Edell, R. McKeown, N. and Varaiya, P. (1995) Billing Users & Pricing for TCP Traffic, *IEEE JSAC*, September,13, 7.
9. Eino, K. and Sakari, L. (2003) Content-based pricing of Services in the Mobile Internet, *proceedings of the 7th IASTED International conference on Internet and Multimedia Systems and Applications (IMSA)*, August, 13-15, Honolulu, Hawaii, USA.
10. NetFlow Services and Applications, Cisco White Paper, Retrieved February 2, from http://www.cisco.com/warp/public/cc/pd/iosw/ioft/neflct/tech/napps_wp.htm
11. The Next Generation Internet Accounting Working Group, Retrieved January 28, from <http://ing.ctit.utwente.nl/WU5/>
12. Sugih, J. B. L. and Shenker, S. (2000) Comments on the Performance of Measurement-Based Admission Control Algorithms, *Proc. of IEEE Infocom 2000*, March 26-30.

13. Kneer, H. Zurfluh, U. Dermler, G. and Stiller B. (2000) A Business Model for Charging and Accounting of Internet Services, *International Conference on Electronic Commerce and Web Technologies (EC-Web 2000)*, September 4-6. Greenwich, U.K.
14. C. Mills, D. Hirsh, G. Ruth, and RFC 1272 - Internet Accounting: Background, Network Working Group, November 1991.
15. Ruth, G. R Usage Accounting for the InternetGTE Laboratories Inc., USA Retrieved February 19, from http://www.isoc.org/inet97/proceedings/F1/F1_1.HTM#Reqts
16. Integrated Services (intserv), the Integrated Service Networking Group, Retrieved January 2, from <http://www.ietf.org/html.charters/intserv-charter.html>
17. The Network Simulator - ns-2, Retrieved December 10, from <http://www.isi.edu/nsnam/ns/>
18. Wireless LAN roaming solutions built upon SS7, HP, Retrieved January 2, from <http://h20208.www2.hp.com/opencall/library/solutions.jsp>
19. Sprenkels, R.A.M. Parhonyi, R. Pras, A. van Beijnum, B.J. De Goede, B.L. (2000) An Architecture for Reverse Charging in the Internet, Proceedings of the IEEE Workshop on IP-oriented Operations and Management IPOM 2000, pages 87 – 92.
20. Monash University, About internet accounting, Retrieved February 10, from <http://www.its.monash.edu.au/i-account/about.html>
21. Varshney, U. Jain, R. (2001) Issues in Emerging 4G Wireless Networks, *IEEE Computer Magazine*, June, 94-96.
22. Zubairi, J. A. El-Shaikh M. A. and Mahmoud, O. (2001) On Shaping and Handling VBR Traffic in a Diffserv Domain, *ATS Simulation Series (ASTC'01)*, 33, 3, April, Seattle, WA, USA.