

Security Issues in a Differentiated Services Internet

A. Striegel

Department of Computer Science and Engineering
University of Notre Dame, USA
striegel@cse.nd.edu

Abstract— The Differentiated Services (DiffServ) architecture represents a highly scalable architecture for deployment of QoS across the next-generation Internet. However, the DiffServ model has several key areas of trust critical to its correct operation which represent security concerns. This paper examines those areas of trust and how they apply to security concerns under the DiffServ model. Finally, the paper examines the proposed solutions to address these concerns.

I. INTRODUCTION

The Internet is continually evolving from a medium used primarily for research and academia to a medium used by business and user communities. For these business and user communities, the applications being used by these communities require different qualities of service (QoS) to be provided to them. However, the Internet in its current form does not support the notion of QoS. Rather, the Internet follows the *same-service-to-all* paradigm such that all packets receive the same quality of service. This best-effort model that Internet currently employs is inadequate to meeting the growing demands of business and user applications.

Therefore, several models have been proposed by the Internet Engineering Task Force (IETF) to provide QoS assurances across the Internet, some of which are likely to be implemented in the next generation Internet. The first model, Integrated Services (IntServ) [1] aims to provide an absolute guarantee of QoS for each flow across the network. This model provides two broad categories of service, *guaranteed service* [2] for real-time flows requiring strict latency and jitter bounds, and *controlled load service* [3] for non-real-time flows which are insensitive to congestion.

The strengths of the IntServ model are that it provides an absolute service guarantee and that it provides a way of monitoring that each flow in the network does not violate its respective allocation of resources. However, the IntServ model has several key weaknesses. Each router requires a significant amount of processing overhead and each router is required to maintain state information for each flow. In the Internet, several million flows may be flowing across a router, thus introducing scalability problems in the IntServ model. In addition, IntServ is impractical for short-lived flows since the connection setup overhead is often greater than the transmission of all packets in a flow. Consequently, the DiffServ model [4], [5] was proposed as an alternative model for providing QoS over the Internet that overcomes the shortcomings of the IntServ model.

The goal of DiffServ was to provide the benefits of different levels of QoS while avoiding the limitations of the IntServ

model. The DiffServ model accomplishes this by aggregating traffic with similar QoS requirements into classes. DiffServ does not maintain any per-flow information across the network, thus eliminating the overhead of maintaining per-flow state information and also eliminating the connection setup costs. Thus, short-lived flows benefit from this model due to the absence of connection setup costs.

II. DIFFERENTIATED SERVICES

Differentiated Services, or DiffServ [4], [5], was the result of addressing of the limitations of IntServ. The goal of DiffServ was to provide the benefits of different levels of QoS while avoiding the limitations of the IntServ model. The DiffServ model accomplishes this by reducing the traffic to some number of aggregations with similar QoS requirements. DiffServ does not maintain any per flow information across the network, thus eliminating the overhead of per flow state information and also eliminating the connection setup costs. DiffServ maintains only information pertaining to the aggregation (classes). Thus, short-lived flows benefit from this model due to the absence of connection setup costs. However, DiffServ does have several weaknesses. Since DiffServ does not employ any admission control or resource reservation, DiffServ adds some measure of unpredictability to the network, thus resulting in extremely dynamic traffic levels. Because of this potential for extremely dynamic traffic levels, DiffServ does not attempt to guarantee any level of service, DiffServ simply strives for a relative level of service for an aggregation versus the other aggregations.

A. Absolute Differentiated Services

With the introduction of the DiffServ model, several variations of DiffServ have been proposed [6], [7], [8]. Absolute differentiated services attempts to provide the levels of performance offered by IntServ without the per-flow states required in the network routers. The user receives an absolute service profile (i.e., a certain bandwidth) from the network. This profile can be of one of two types of services, Premium Service or Assured Service. The first, Premium Service [6] is equivalent to a leased line provided that service stays below a certain level. The second, Assured Service, classifies packets into two levels regarding their drop preference, 'In' or 'Out'. 'Out' packets are discarded with a higher probability than 'In' packets during network congestion [7]. A third, Assured Forwarding [8] expands the levels of drop precedence into three levels beneath four main classes. In the absolute differentiated services model, there are tradeoffs between achieving high service assurance

versus coarse spatial granularity (certain bandwidth in many or even all network paths) which is discussed in [9].

B. Relative Differentiated Services

The second variation of DiffServ is relative differentiated services. For relative differentiated services, all traffic is grouped into N classes of service. For each class i , the service provided to class i will be better (or at least no worse) than the service provided to class $(i-1)$, where $1 < i \leq N$, in terms of queuing delays and packet losses [10]. The “or at least no worse” clause is included for levels of low network activity where all classes experience the same quality of service. An application selects its Class Per-Hop Behavior (PHB) as defined by the IETF [5] to select the appropriate level of service for the application. However, this level of service is relative to the other classes in the network and is not an absolute guarantee. The relative differentiated services model assures that the performance of the selected class will be relatively better than the performance of lower classes in the network.

To implement relative differentiated services, several approaches have been proposed. The first approach, price differentiation, proposed by the Paris Metro Pricing (PMP) scheme [11], uses only pricing of classes to differentiate services with the assumption that higher pricing will lead to lower loads and hence lower loads in the higher classes. A second approach, *careful capacity provisioning*, involves higher classes having more forwarding resources relative to their expected loads through the use of schedulers such as Weighted Fair Queuing [12], [13], [14]. However, each of these two approaches has the same problem when dealing with Internet traffic. Because of the bursty nature of Internet traffic, a higher class may be overloaded, thus performing worse than a lower class [15], [10]. The third approach, strict prioritization, provides consistent class differentiation that does not depend on load. Strict prioritization accomplishes this by servicing the highest backlogged class (delay aspect) and drops a packet from the lowest backlogged class (loss aspect). However, strict prioritization presents two problems. First, lower classes can experience starvation effects if no restriction is placed upon the load of the higher classes. Second, strict prioritization does not provide for controllable differentiation between the classes.

Recently, the proportional differential model was proposed in [10]. In the proportional differential model, QoS performance measures are ratioed proportionally via the use of class differentiation parameters. The proportional differential model was developed with two criteria in mind for successful service differentiation. First, a model must be *predictable*, such that the differentiation is consistent (a higher class is better or at least no worse than a lower class) and the differentiation is independent of class loads. Second, the model must be *controllable*, such that the network operators can select the appropriate level of spacing between the classes based on their criteria. Although [16] addressed only delay differentiation, several papers [10], [17] have addressed both delay and loss differentiation, two critical metrics for QoS across the Internet.

All of the proposed variations of DiffServ follow several key concepts which are discussed in Section III. DiffServ is expected to become a dominant force in the Internet and because

of this, the security concerns must be addressed. Section IV outlines the areas of trust in the DiffServ model, the potential areas of concern, and the proposed solutions. Finally, in Section V, some concluding remarks are made.

III. DIFFSERV CONCEPTS

In order to maintain compatibility with the existing IPv4 infrastructure and IPv6, DiffServ represents a relatively minor change to the actual IP packet. Rather than incorporating an actual design change, DiffServ incorporates only a semantic change by redefining the use of the TOS field. The newly renamed field, Differentiated Services (DS) field, uses the first 6 bits of the TOS field while the remaining 2 bits are reserved for future use.

Each value in the DS field, known as a DiffServ codepoint (DSCP) [4], is responsible for aggregating packets into classes. Each different class is associated with a specific Per-Hop Behavior (PHB) [5] which defines how a packet will be prioritized for transmission and dropping due to buffer overflow.

The DiffServ model contains two types of routers, *edge routers* and *core routers*. Core routers are relatively simple routers designed for the purpose of high-speed routing over the network backbone. Core routers do not maintain any per-flow state information and schedule the packets as per the PHB defined within each packet.

In DiffServ, the intelligence of the network is migrated to the edge of the network at the edge routers. The edge router is a critical key to the correct operation of the DiffServ network. Responsibilities of the edge routers include proper marking of non-DiffServ-aware traffic, traffic policing, and traffic shaping. It is these edge router responsibilities that are responsible for maintaining proper traffic levels to achieve QoS differentiation in the network core.

If a network sending traffic to the DiffServ domain is a DiffServ network, the traffic is already marked and thus only needs to be policed. However, if a network is not DiffServ-aware, the edge router must be responsible for appropriately marking packets for the DiffServ domain. The packets are marked according to a Service Level Agreement (SLA) between the source and edge router. A SLA exists between an edge router and a source to outline limitations for each class of service in regards to both the quantity of traffic as well as the burstiness of the traffic. For traffic that violates a SLA, the offending packets may either be demoted to a lower class of service or dropped. A SLA may be either static or dynamic.

The DiffServ model is shown in Figure 1. A LAN or MLAN transmits its packets to an ISP’s edge router. If the LAN is not DiffServ-aware, the packet is appropriately marked according to the SLA between the LAN and ISP. The traffic from the LAN is policed according to the SLA. Packets are then scheduled/dropped on the DiffServ domain according to the PHB in the packet.

IV. SECURITY AND DIFFSERV

A. Areas of Trust

Several areas of trust exist in the DiffServ network which are fundamental to the correct operation of DiffServ. These

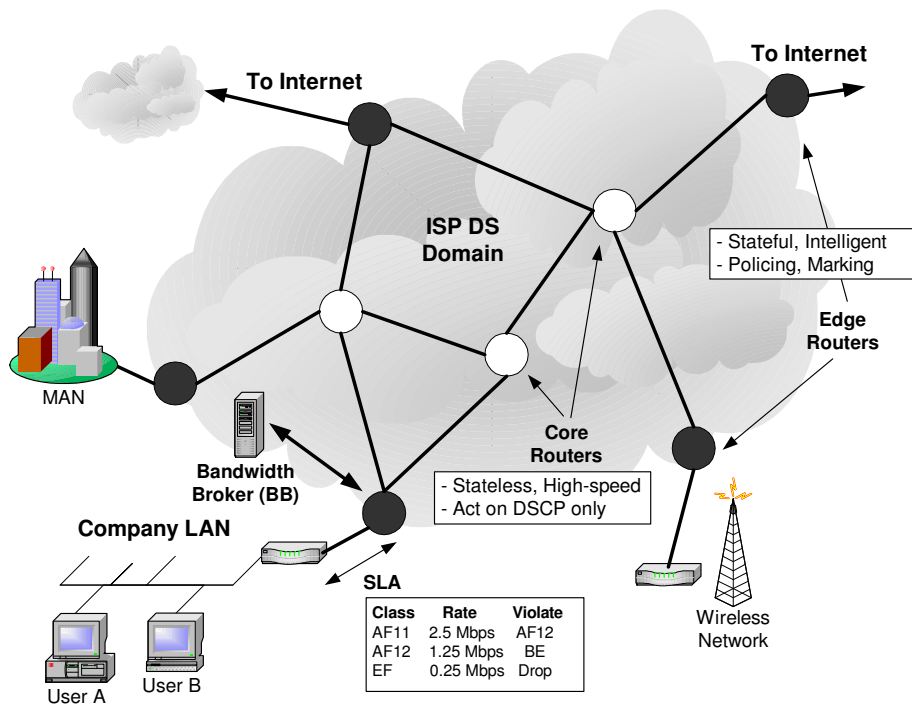


Fig. 1. DiffServ model

areas of trust include trust between edge router and source, trust between core and edge routers, and trust of SLA integrity.

1) *Trust between Edge Router and Source:* Packets are policed on a per-source basis at the edge router. Thus, in order to police a packet or mark a packet, the source of a packet must be matched to an SLA in the edge router. Source to SLA matching can occur on either the physical layer (less likely) or at the network layer (likely). The edge routers trust that the source to SLA matching is done correctly in order to correctly police traffic.

2) *Trust between Core and Edge Router:* The primary goal of DiffServ is to simplify the core routers to allow for high-speed routing of packets according to the PHBs of the packets. Thus, the core routers have a level of trust with the edge routers such that the core routers trust that the packets have been marked correctly and also trust that the packets have already been appropriately policed.

3) *Trust of SLA Integrity:* Several services such as Expedited Forwarding [6] and Assured Forwarding [8] depend on SLA integrity in order for them to function correctly. If the classes are overloaded with excessive traffic, performance to lower classes or even the performance of the higher priority classes could degrade. Thus, a level of trust exists with the integrity of SLAs across edge routers such that the network resources are not overallocated to cause performance degradation of stricter QoS classes.

B. Potential Security Concerns

The areas of trust that are critical to the DiffServ model represent several potential security concerns. These security concerns include both theft of resources as well as Denial of Service (DoS) attacks.

1) *Theft of Resources:* Theft of resources can occur in several forms under DiffServ. Theft in terms of DiffServ can be expanded to include theft of network bandwidth as well as illegal promotion of packet's PHB. The first, theft of bandwidth, can occur at both the edge router and core router level. At the edge router level, if a packet is able to successfully spoof its source, the packet will have stolen part of the actual source's SLA allocated bandwidth. Theft of bandwidth at the core router level can occur if an edge router transmits traffic beyond the SLAs or traffic bypasses edge routers is transmitted directly onto the core.

The second type of theft, illegal promotion of a packet's PHB can occur at both the edge and core router. At the edge router, illegal promotion can occur if a packet is policed incorrectly or not at all. At the core router, illegal promotion can occur if the correct PHB behavior is not enforced, i.e. a rogue core router or malfunctioning core router.

2) *Denial of Service:* Denial of Service in the context of DiffServ represents a complete theft of resources over the DiffServ network. Denial of Service is a major security risk to DiffServ and can occur on several fronts.

First, a Denial of Service attack can occur at the edge router with outgoing traffic. The policing of flows represents an attack point that can be exploited to issue a Denial of Service attack. Because the edge router polices on a per-source basis, a simple Denial of Service attack would be to flood the edge router with a spoofed source in order to penalize legitimate traffic arising from the source. This requires only knowledge of the SLA to source matching methodology being employed at the edge router (physical or network layer matching).

A second point for a Denial of Service attack can occur again at the edge router. However, in this case the edge router refers

to the edge router at the edge of the ISP's network to other domains. As with the edge routers to the LANs connected to the ISP, the ISP also maintains an SLA with other domains at the edge of its network. Thus, it would be possible to conduct a Denial of Service attack from either inside the ISP's network for outgoing traffic or outside of the ISP's network for incoming traffic by overloading the edge router to violate the SLA and cause excessive penalization of the target's packets. This attack requires knowledge of the network infrastructure.

The third attack point for DoS occurs within the core routers themselves and is rooted with the SLAs for the network. By overloading a class over the network, it is possible to cause the class to experience much worse performance and even adversely affect traffic from other classes as well, denying the service differentiation normally offered by DiffServ. This can occur due to either an over-allocation of SLAs at the edge routers or due to excessive congestion around specific core routers. The first requires bypassing of the edge router policing while the second requires knowledge of then network infrastructure.

C. Proposed Solutions

As a result of these potential security concerns, the IETF DiffServ working group has outlined several methods for use with DiffServ in order to address those concerns. Currently, the Architecture RFC [4] considers only auditing and IPSec.

1) *Auditing*: Auditing is included as a way to monitor suspicious events in the DiffServ domain. Auditing is not required as part of a DiffServ domain but is recommended when included in a system (overall framework) that supports auditing. An example of an auditable event would be traffic on an unused code-point at a core router. Auditing can be used to increase both the security and robustness of the network. However, to avoid a potential DoS attack, there is no requirement at any time for a node that detects an auditable event to transmit a message to the purported sender.

2) *IPSec*: IPSec, outlined in [18], [19], is an extension to IP to allow for secure IP based transmission. In its default mode, IPSec does not include the DS field in its cryptographic calculation. Thus, the default mode is not suited for providing security to DiffServ domains. However, IPSec tunnel mode does provide security that is of direct use to a DiffServ domain. Tunnel mode includes two versions of the IP header, an inner encrypted version of the header and an outer version used for transmission. However as with default mode, the outer IP header still is not included in the cryptographic calculation, thus rendering it vulnerable to a *man-in-the-middle attack*.

In order to use IPSec's tunnel mode, several points must be considered. First, the core routers examine only the outer IP header. The inner IP header can only be examined at either the ingress or egress node of the domain. The ingress node can use IPSec to correctly match the source to its appropriate SLA while the egress node can use IPSec to check the end-to-end integrity of the packet. The security of this scheme is dependent upon the strength of the integrity check used.

A final point to consider arises at the egress node. As it stands currently, the egress node between DiffServ domains is not allowed to modify the inner DS field in order to apply traffic conditioning. However, if modification is allowed, it increases net-

work adaptiveness at the cost of security. Thus, the egress node between two DiffServ domains must now include the appropriate security found in an ingress node, thus greatly increasing the complexity of the nodes between DiffServ domains. Essentially, the network may be viewed either as a 'virtual wire' with no inner DS field modification or as a multihop network which allows inner DS field modification.

V. CONCLUSION

In conclusion, the DiffServ architecture represents a highly scalable architecture for deployment of QoS across the next-generation Internet. However, the DiffServ model has several key areas of trust which represent security concerns critical to the correct operation of DiffServ. These concerns have been addressed by the IETF in DiffServ Architecture RFC but there is still room for investigation into DiffServ security.

REFERENCES

- [1] R.Braden, D. Clark, and S. Shenkar, "Integrated Services in the Internet architecture: An overview," *IETF RFC 1633*, June 1994.
- [2] S. Shenkar, C. Partridge, , and R. Guerin, "Specification of Guaranteed Quality of Service," *IETF RFC 2212*, Sept. 1997.
- [3] J. Wroclawski, "Specification of the controlled-load network element service," *IETF RFC 2211*, Sept. 1997.
- [4] K. Nichols, S. Blake, F. Baker, and D.L. Black, "Definition of the Differentiated Services field (DS Field) in the IPv4 and IPv6 headers," *IETF RFC 2474*, Dec. 1998.
- [5] S. Blake et. al, "An Architecture for Differentiated Services," *IETF RFC 2475*, Dec. 1998.
- [6] B. Davie, A. Charny, J.C.R. Bennet, K. Benson, J.Y. Le Boudec, W. Courtney, S. Davari, V. Firoiu, and D. Stiliadis, "An expedited forwarding PHB (per-hop behavior)," *IETF RFC 3246*, Mar. 2002.
- [7] D.D. Clark and W. Farang, "Explicit allocation of best effort packet delivery service," *IEEE/ACM Transactions on Networking*, vol. 6, pp. 362–373, Aug. 1998.
- [8] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski, "Assured forwarding PHB group," *IETF RFC 2597*, June 1999.
- [9] I. Stoica and H. Zhang, "LIRA: An approach for service differentiation on the internet," in *Proc. of NOSS-DAV*, 1998.
- [10] C. Dovrolis and P. Ramanathan, "A case for relative differentiated services and the proportional differentiation model," *IEEE Network*, pp. 26–34, Sept.-Oct. 1999.
- [11] A.M. Odlyzko, "Paris metro pricing: The minimalist Differentiated Services solution," in *Proc. IEEE/IFIP International Workshop on Quality of Service (IWQoS)*, June 1999.
- [12] A. Demers, S. Keshav, and S. Shenker, *Analysis and Simulation of a Fair Queuing Algorithm*, 1990.
- [13] A.K. Parekh and R.G. Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single-Node Case," *IEEE/ACM Transactions on Networking*, vol. 1, pp. 344–357, June 1993.
- [14] J.C.R. Bennett and H. Zhang, "Hierarchical Packet Fair Queuing Algorithms," *IEEE/ACM Transactions on Networking*, vol. 5, pp. 675–689, Oct. 1997.
- [15] D. Stiliadis and A. Varma, "Latency-rate servers: A general model for analysis of traffic scheduling algorithms," *IEEE/ACM Transactions on Networking*, vol. 6, pp. 611–625, Oct. 1998.
- [16] C. Dovrolis, D. Stiliadis, and P. Ramanathan, "Proportional Differentiated Services: Delay Differentiation and Packet Scheduling," in *SIGCOMM*, 1999, pp. 109–120.
- [17] A. Striegel and G. Manimaran, "Packet scheduling with delay and loss differentiation," *Computer Communications*, vol. 25, no. 1, pp. 21–31, Jan. 2002.
- [18] S. Kent and R. Atkinson, "IP Encapsulating Security Payload (ESP)," *IETF RFC 2406*, Nov. 1998.
- [19] R. Atkinson, "IP Authentication Header," *IETF RFC 1826*, Aug. 1995.