Session Classification for Traffic Aggregation

Hui-Kai Su, Cheng-Shong Wu and Kim-Joan Chen
Department of Electrical Engineering
National Chung-Cheng University
Chia-Yi, Taiwan 621, ROC
Email: pat@ee.ccu.edu.tw, {ieecsw, ieekjc}@ccu.edu.tw

***Abstract*— Classification is a critical task to be performed for network switches and routers to guarantee Quality of Service. Some applications can be categorized efficiently by using the multi-field classifiers and content-aware classifiers. However, this traditional classifications are inadequate for those applications whose classification rules can't be configured in advance. Most of this kind applications are session based, that is more than one transport channels are needed in their communication. In this paper, we propose the state tracking session classification and present two implementation alternatives. An implantation example of session classification for H.323 VoIP is also presented in this paper. By tracking the states of the sessions, all of the packets in the session-based applications can be caught and then marked or tagged. In addition, the proposed session classification can cooperate with different aggregation techniques such as Diffserv, MPLS, and VPN to achieve efficient QoS control in a network.**

I. Introduction

Today, not only packet forwarding and routing, but also quality of service (QoS) guarantee and traffic engineering are necessary to be performed in switches or routers. Classification is a critical task for providing Inserv/Diffserv (Integrated/Differentiated Services) [1] [2] [3], supporting traffic engineering and realizing variant traffic aggregation technologies, such as MPLS (Multi Protocol Label Switching) [4] [5], VPLS (Virtual Private LAN Service) [6]. In an ingress node of QoS control domain, the incoming packets are categorized by classifier and then marked with a label or added a tag, which indicates the routing path, the service level or the virtual network ID. Multiple data flows can be exactly classified and aggregated into a virtual path, which is preset-up with traffic engineering, and/or into predefined service classes to achieve the class-based QoS guarantee.

The traditional classifications are divided into two types, multi-field classification and content-aware classification. The classifications to be performed encompass a wide range, from well-understood operations such as route table lookups to complex packet identification involving multi fields in the packet. The multi-field classification based on packet-filtering algorithms have been studied in [7], [8], and [9]. Furthermore, the advantages of content-aware network devices demand the use of flexile packet classifiers that can handle operations such as pattern searches and regular expression matching [10].

To date, theses classification devices, such as CAMs/ternary CAMs (TCAMs), special function ASICs, and software-based algorithms have been deployed. However, these solutions are inadequate for most session-based multimedia applications listed in Tab. I. In fact, these session based applications are more desperate for QoS guarantee than others. Two types of channels, the control channel and the data channel, are used for communication is the main feature of these applications. The control channel is used to exchange control messages to set up data channel; while and the multimedia data are transmitted over data channel. Usually, the control channel is established on a default or well-known TCP/UDP port (Transport address). However, the data channel may be set up on a dynamic or unknown TCP/UDP port which is decided during connection negotiation by the exchange of control message.

For example, Windows Media Player is used to play broadcast live or on-demand Windows Media clips. There are three modes, which are over-TCP, over-UDP and over-HTTP modes, for the data channels in this player. For over-TCP mode, MMS (Microsoft Media Services) protocol is used and the multi-filed classification can be used since both the control and data channels use pre-determine TCP or UDP port. For over-HTTP mode, we need look into the content of HTTP data to determine the transmitted application. Only content-aware classification can achieve the job. However, for over-UDP mode, both multi-field classification and content-aware classification fail to work because the data channel is unknown in advance. Therefore, we need new technique to classify this kind of session based multimedia application. Please note that in Tab. I applications/protocols with "$+$" maker are the applications that can be classified by multi-field classifier, applications/protocols with "$\times$" maker can be classified by content-aware classifier, and those with "$*$" need our new proposed session based classifier. In any case, the session classification, the multi-field classification and the application-aware classification have to be supported simultaneously, if we want to provide QoS guarantee for all Windows Media Player applications.

This paper proposes the state tracking session classification and presents two implementation architectures. The basic concept of our state-tracking session classification is to monitor the control channel to learn the Transport address of data channel, and set the classifier dynamically to classify the data traffic.

In the following section, we first discuss the session classification architectures. In Section III, a session classification example for H.323 VoIP applications is presented. In Section IV, we discuss some features of the session classification, and make concluding remarks.

TABLE I

PORT USAGE IN SESSION-BASED APPLICATIONS

Software	Application Protocol	Control Channel	Data Channel
Windows Media Player	MMS^{+}	1755/TCP	1755/TCP
	*MMS**	1755/TCP	1024-5000/UDP or 1-65000/Multicast
	HTTP$^{\times}$	80/TCP	80/TCP
RealOne Player, RealPlayer, and QuickTime Player	RTSP^{+}	554/TCP	554/TCP
	*RTSP**	554/TCP	6970-32000/UDP
	HTTP$^{\times}$	80/TCP	80/TCP
RealPlayer 5 and earlier version	PNA^{+}	7070/TCP	7070/TCP
	*PNA**	7070/TCP	6970-32000/UDP
	HTTP$^{\times}$	80/TCP	80/TCP
FTP	FTP^{+} (active)	21/TCP	20/TCP
	*FTP * (passive)*	21/TCP	Dynamic/TCP
H.323 IP Phone	*H.323**	1720/TCP, and dynamic/TCP	Dynamic/UDP
SIP IP Phone	*SIP**	5060/UDP	Dynamic/UDP

II. SESSION CLASSIFICATION ARCHITECTURES

A. Traditional Classification Architecture in a Network Device

The system architecture of a typical network device, such as router, or switch, that supports QoS is showed in Fig. 1. Functionally, the device consists of management plane, data plane, and control plane. Management functions are implemented in management plane, e.g. routing information management, and classification rule management, and etc. Control functions and routing protocols belong to control plane, e.g. OSPF, RIP, and etc. Furthermore, packets processing and forwarding are designed in data plane, e.g. traffic conditioner, routing/switching, scheduling, and etc.

Fig. 1 illustrates a classification function in a network device. Classification rules are predictably configured by administrators, using console mode configuration or network management protocol, e.g. SNMP (Simple Network Management Protocol), or COPS (Common Open Policy Service). Moreover, they are maintained in management plane and set into classifier in data plane. According to these classification rules, incoming packets will be classified by using multi-filed classification or content-aware classification and marked with a label or a tag. After that, the processes of the packets, including traffic condition, buffer management and scheduling, are all based on the labels or the tags.

Because of dynamic allocation of the data channel during communication, the classification rules of the data channels in session-based applications are unknown in advance. The traditional classifications which configured by management plane are inadequate for these applications. Thus, the state tracking section classification is proposed in this paper. This ideal behind the session classification is straight forward. First, the session sate is traced ,and the data channel is figured out. Second, the classification rules of these channels are configured into classifier. Therefore, this solution can solve the classification issue for session-based applications effectively.

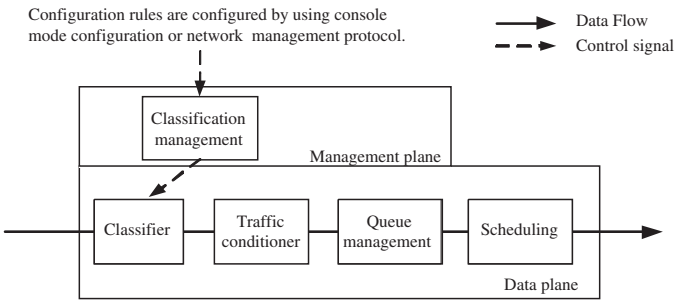

Fig. 1. A classification function in a network device.

B. State Tracking Session Classification

The state tracking session classification proposed in this paper are supplementary to support the traditional classifier to figure out all the channels of session-based applications that are contracted with the SLA (Service Level Agreement) [2]. It handles the session tracking function that tracks session state in each session-based application to find out all of the communication channels used by the application. Furthermore, the classification rules are translated and configured into the traditional classifier automatically. Based on this procedure, the packets of session-based applications can be classifiable. In section III, we will illustrate the details of state-tracking session-based classification by the example of H.323 VoIP applications.

C. The Implement of Session Classification

Here, we propose two implement alternative of session classification. One is called internal session classification, and the other is called external session classification, illustrated in Fig. 2 and Fig. 3 respectively. The common function of these two architectures is that the session state tracker that performs session state monitoring and information management. Then, the classifier will classify incoming packets based on the classification rules that are translated from session state information and configured by the tracker.

In the internal session classification architecture (Fig. 2), the session state tracker is added between the management plane and the control plane. After gathering the data channel information from the control channel, the state tracker translates the classification rules and request the traditional classification management module to set up the classifier immediately. This implementation is approachable by modifying the present network device software, e.g. IOS (Internet Operation System). However, the processing power and memory size of network devices must be considered.

In the consideration of the processing power and the flexibility of network devices, the external session classification architecture (Fig. 3) is implemented by added an external session classification agent. The advantages of this architecture is easy to provide session classification in the traditional network devices. However, the agent is necessary to be installed in each ingress port of the network devices in an QoS domain. The cost of the external session classification architecture may be

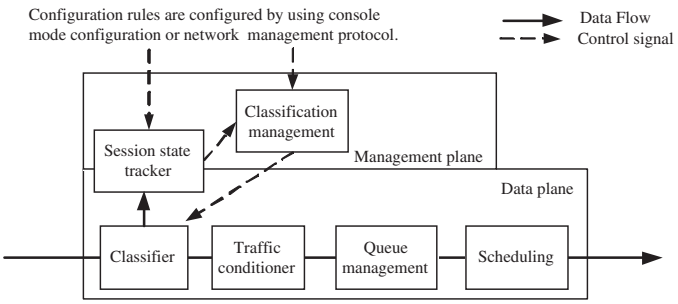

Fig. 2. Internal session classification architecture.

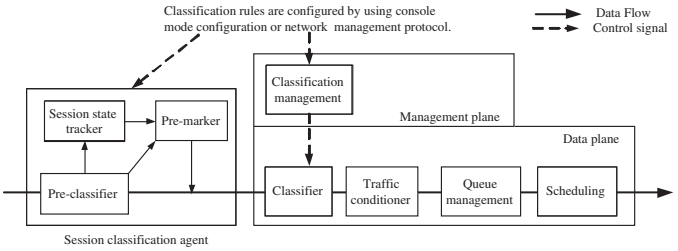

Fig. 3. External session classification architecture.

higher, but the performance should be better than the other implementation.

In our experiment, the external architecture is adopted and the session classification agent is designed and implemented on a Personal Computer platform. The functional blocks of this agent are illustrated in the left of Fig. 3. Linux operation system is installed and working with bridge mode. Each packet is preclassified while received. First, if the packet filtered by the pre-classifier carries a control message on the control channel of a session-based application which belongs to a customer with SLA contract, it will be copied into the session state tracker to abstract the session state information. Then, it will be delivered to the ingress router with a suitable label or tag that identifies its service level or forwarding path. In addition, according to the information from control packet, the pre-classifier will be configured to catch the user data on the data channel. Second, if the packet carries user data on the data channel of the session-based application, it will be marked with the label or tag and will be delivered to the ingress router. For all of the other packets, they are not necessary to be processed in the agent and will be deliver to the ingress router directly. However, in the opposite direction, the packets is to leave the QoS domain, so they bypass the agent.

III. A Session Classification Example for H.323 VoIP Applications

A. H.323 Overview

The session structure of H.323 VoIP applications are the most complex among all session-based applications mentioned in Tab. I. It is constructed by three different control channels and data channel. The control channels contain H.225 call control channel, H.245 control channel and H.245 media control channel (use RTCP, Real-Time Control Protocol), and

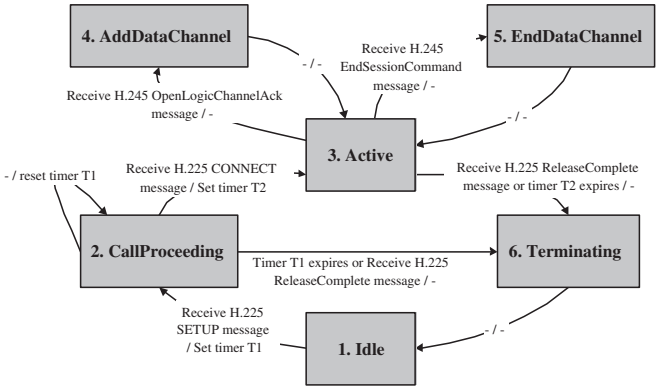

Fig. 4. Finite state machine in session state tracker.

the data channel is H.245 media channel (use RTP, Real-Time Protocol) [11] [12].

The procedure of H.323 session, shown in Fig. 5, contains five phases: the call setup, the initial communication and capability exchange, the establishment of audiovisual communication, the call services, and the call termination phases. Note that, the five phases of H.323 section are denoted as phase A, B, C, D and E respectively in Fig. 5. In the call setup phase, a H.225 call control channel is established on the well-known port (1720/TCP). The H.245 control channel is negotiated in this phase and will be used in the next two phases. When the call is picked up by the callee, the session enters the initial communication and capability exchange phase. H.245 media control channel and H.245 media channel are resolved in the establishment of audiovisual communication phase. After conversation, all of these control channels and data channels are released and it enters the call termination phase.

B. Design of H.323 Session Classifier

In order to classify each packet for H.323 VoIP applications, all information of the control channels and the data channels has to be found out and managed. The finite state machine in the session sate tracker is designed and illustrated in Fig. 4.

In a beginning, the session state is initiated in Idle state. When H.225 SETUP message is detected, the session sate will change to CallProceeding state and timer $T1$ is set. In this state, the H.225 call control channel information will be caught and analyzed in the successive packets. In order to prevent management errors, the timer $T1$ is necessary. If establishment time of call connection is larger than $T1$, the call will be aborted. In CallProceeding sate, if H.225 ReleaseComplete message is detected or timer $T1$ is expired, the session will move to Terminating sate. If this call is picked up by the callee and H.225 CONNECT message is detected, the session state will change to Active state and timer $T2$ will be set. H.245 control channel will be analyzed. In other words, we assume that the maximum conversation time in this system is $T2$. If this call are not released within timer $T2$, this call will be terminated. If other H.225 messages is detected, timer $T1$ will be reset. In Active state, H.245 media control channel and H.245 media channel are dynamically established and

released according to the results negotiated in H.245 control channel. When H.245 OpenLogicChannelAck message is detected, the H.245 media control channel information and the H.245 media channel information will be caught and managed. When H.245 EndSessionCommand message is detected, the information will be removed from the tracker. Finally, if H.225 ReleaseComplete message is detected or timer $T2$ is expired, the session sate will change to Terminating sate, and the all the information of this session will be removed.

When the channel information of the control channels and data channels is caught by the tracker, it is translated into classification rules and configured in the pre-classifier automatically. On the other hand, when this call is released, these classification rules and all of the data about the session are removed from the pre-classifier automatically.

The directions of the H.245 media control channel and H.245 media channel are opposite, and the only ingress direction of these channel information is useful for the pre-classifier. Thus, the direction has to be distinguished.

C. Experiment Environment

Our experiments on the session based classification is part of our Diffserv/MPLS project. Our project is to study, design, implement ,and validate a set of application QoS oriented service differentiation and traffic engineering over MPLS. The Diffserv/MPLS network devices are designed according to [5] on IXP1200 (Intel Network Processor) platforms. In the consideration of the flexibility and processing power of our platform, we use the external session classification architecture and design a session classification agent supplemented to ingress port of the Diffserv/MPLS edge for session-based VoIP applications.

Diffserv provides class-based QoS guarantees, and achieves scalability by aggregating traffic classification. Packets are classified and marked to receive a particular per-hop forwarding behavior on nodes along their path. Sophisticated classification, marking, policing, and shaping operations need only be implemented at network boundaries or hosts. The core routers only process packets based on a small number of Per-Hop Behaviors (PHB) encoded in the packet header [2] [3]. MPLS provides fast forwarding and traffic engineering by using label swapping technologies between Data Link Layer and Network Layer. It avoids independently choosing a next hop for the packet, based on its analysis of the packet's header and the results of running the routing algorithm [4].

In this Environment, each packet of the session-based applications is premarked by the agent with a DSCP (DiffServ CodePoint) in the IP header to indicate the Diffserv service class that the application belongs to. Then, the edge routers only process packets based on the PHB of this DSCP. Besides, the other packets are transparently bypassed in the agent and enter into edge routers.

The management information of our implementation is showed in Tab.II. The SLA examples for VoIP applications are listed in Tab. II-a which includes a set

TABLE II

INFORMATION MANAGEMENT IN SESSION STATE TRACKER

a. Network level contract for service state keeping agent

SA	DA	DSCP	AP
140.123.107.0/24	140.111.1.0/24	101110	VoIP
140.123.109.0/24	140.123.106.0/24	101110	VoIP

b. Control channel table

S_ID	Direction	SA	DA	SP	DP	PID	DSCP	AP	State	Timer
1	0	140.123.107.171	140.111.1.100	3867	1720	TCP	101110	VoIP	ACTIVE	3000
2	0	140.123.108.10	140.123.109.30	3894	1720	TCP	101110	VoIP	INIT	30

Direction: 0: Caller to Called, and 1: Called to Caller

c. Logical channel table

DC_ID	PDC_ID	S_ID	SP	DP	PID	Type
1	0	1	3868	3931	TCP	0
2	1	1	5004	5004	UDP	1
3	1	1	5005	5005	UDP	2

Type: 0: H.245 control channel,
1: Media channel for voice, and
2: Media control channel for voice

of source addresses and destination addresses, application types, and DSCPs. In this example, the traffic of VoIP from 140.123.107.0/24 to 140.111.1.0/24 and from 140.123.109.0/24 to 140.123.106.0/24 is to be provided the QoS guarantee of the EF (Expedited Forwarding) class in the Diffserv domain.

H.225 call control channel information is managed and listed in Tab. II-b. The management information of H.225 call control channel consists of Session ID (S_ID), Call Direction (Direction), Source IP Address (SA), Destination IP Address (DA), Source Port (SP), Destination Port (DP), Protocol ID (PID), DiffServ CodePoint (DSCP), Application Type (AP), Session State (State), and Timer. In this example , according to the above SLAs, when the session state tracker is tracing a call from 140.123.107.171 to 140.111.1.100, the H.225 call control channel information is filled into the row with S_ID 1 in Tab. II-b.

Furthermore, H.245 channel information, including H.245 control channel, H.245 media control channel and H.245 media channel, is listed in Tab. II-c. The management information of H.245 channel includes Data Channel ID (DC_ID), Parent of Data Channel ID (PDC_ID), Session ID (S_ID), Source Port (SP), Destination Port (DP), Protocol ID (PID) and Channel Type (Type). The S_ID is related to the S_ID of Tab. II-b. Following the previous example, all of the H.245 channels are managed as DC_ID 1, 2, and 3 in Tab. II-c. Finally, the information of Tab. II-b and Tab. II-c is translated into classification rules to configure the pre-classifier by event driven.

The call procedure and the translation of session states in the session classification agents are illustrated in Fig. 5. These two agents on the two edge of Diffserv Domain works independently on tracking the sates of this session and classify packets. After Phase A, the session states of them change to Active state, and the H.245 control channel is caught by each agent. After Phase C, the H.245 media control channel and H.245 media channel of the ingress direction are caught, and all channel information in this session is translated into classification rules and configured into the pre-classifier automatically. Finally, all of these channels are released in Phase

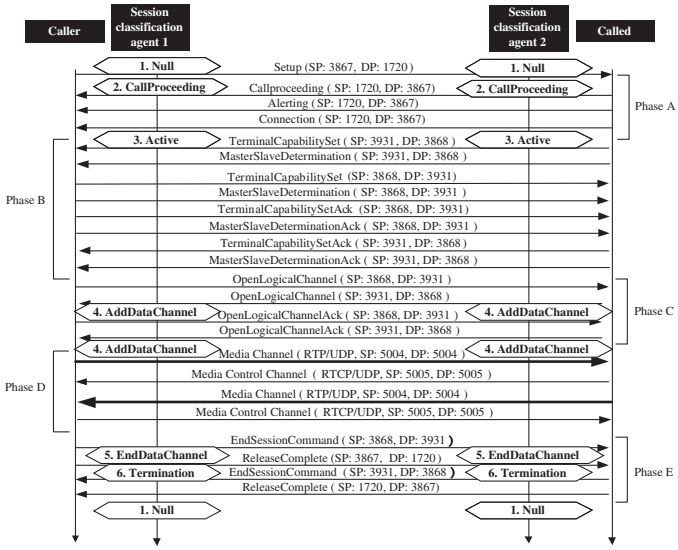

Fig. 5. A message sequence chart with H.323 VoIP applications.

E, and all the information and the classification rules related this session are removed from the pre-classifier automatically. Therefore, in the ingress network device, each VoIP packet is premarked with the DSCP code that indicates which Diffserv service level it belongs to in our example.

Based on the same concept, session classifier is easy to extend to other session-based applications. It is also ready for other traffic aggregation technologies such as MPLS and VPN. For example, if MPLS is used, instead of mark DSCP code, a shim header will be added to indicate which LSP the session belongs to. Although we realize the session classification on PC, the agent could be condensed as a module device to plug into the network interfaces of traditional network devices.

IV. Conclusion

The state tracking session classification is proposed in this paper. In the consideration of the processing power, the flexibility and the cost of network devices, the internal session classification architecture and the external session classification architecture are discussed. In our experiment, the session classification agent for H.323 VoIP applications is realized in a PC platform. The session classification is easy to extend to other session-based applications and to apply to MPLS, VPN, and other traffic aggregation technologies.

References

[1] R. Braden, D. Clark, and S. Shenker, "Integrated Services in the InternetArchitecture: an Overview," RFC 1633, June 1994.

[2] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An Architecture for Differentiated Services," RFC 2475, Dec. 1998.

[3] D. Grossman, "New Terminology and Clarifications for Diffserv," RFC 3260, Apr. 2002.

[4] E. Rosen, A. Viswanathan, and R. Callon, "Mutilprotocol Label Switching Architecture," RFC 3031, Jan. 2001.

[5] F. L. Faucheur, L. Wu, B. Davie, S. Davari, P. Vaananen, R. Krishnan, P. Cheval, and J. Heinanen, "Multi-Protocol Label Switching (MPLS) Support of Differentiated Services," RFC 3270, May 2002.

[6] K. Kompella and Y. Rekhter, "Virtual Private LAN Service," Internet Draft, July 2004.

[7] M. Uga and K. Shiomoto, "High speed policy based packet forwarding using efficient multi-dimensinal range maching," in *Proc. ACM SIGCOMM*, Vancouver, Canada, Sept. 1998.

[8] P. Gupta and N. McKeown, "Algorithms for packet classification," *IEEE Network*, vol. 15, pp. 24–32, March/April 2001.

[9] M. Uga and K. Shiomoto, "A modular approach to packet classification: Algorithms and results," in *Proc. INFOCOM*, Israel, Mar. 200.

[10] S. Iyer, R. R. Kompella, and A. Shelat, "ClassiPI: An Architecture for Fast and Flexible Packet Classification," *IEEE Network*, vol. 15, pp. 33–41, March/April 2001.

[11] B. Goode, "Voice over Internet protocol (VoIP)," *Proceedings of the IEEE*, vol. 90, pp. 1495–1517, Sept. 2002.

[12] *H.323v4: Packet-based multimedia communications systems*, ITU-T Std., Nov. 2000.