Performance analysis of multi-class Internet traffic classifier in a connection oriented router environment

Mika Ilvesmäki, Marko Luoma

Helsinki University of Technology
Laboratory of Telecommunications Technology
Otakaari 5 A
ESPOO, Finland

ABSTRACT

In this work, we analyze the performance of multi-class Internet traffic classifier primarily in a connection-oriented IP router environment. We define the tasks and related concepts of traffic classification in the Internet and then proceed to construct a multi-class traffic classifier using the Learning Vector Quantization algorithm classifier that has been previously used to divide the traffic into two classes. We show how the functionality of the 2-class LVQ classifier can easily be extended to an arbitrary amount of classes, in this work to three: the hard-interactive, the elastic and the best effort service classes. This is done by consecutive 2-class classifications and unanimous decisions on classification. The multi-class LVQ classifier is observed to perform adequately being able to provide network service profiles, sets of classified services, with characteristics relating to respective classes. The multi-class classifier also performs well within the hardware performance limits and is still able to provide the forwarding component of the Internet router a substantial workload reduction. We also observe the individual service class performance and conclude that separately defined service architectures are needed to limit the use of QoS resources in the network. Our belief is that in order to provide end-to-end QoS a connection oriented link layer technology is needed. These technologies are able to offer dedicated forwarding paths for individual packets thus reducing the processor intensive hop count and flattening the network.

Keywords: Traffic classification, Internet, Router Performance, Quality of Service, Artificial Intelligence

1. INTRODUCTION

Some applications used in the Internet might function adequately without any particular service level while others could substantially benefit if they were offered higher levels of service. Several differing solutions and architectural suggestions have been presented and studied to introduce service levels and prioritization to the Internet. Service architectures like Differentiated Services[1,2] and Integrated Services[3,4] aim to provide varying levels of either absolute or relative quality to the critical traffic in the Internet. In conjunction, new architectural concepts have also emerged to offer the Internet routers enhanced performance through fast packet forwarding,[5] fast address look-ups[6] or forwarding workload reduction[7] by redirecting traffic flows to the connection layer away from the actual routing and forwarding processes in the router.

The growing pressure on the performance of the Internet protocol -based networks requires dealing with several issues such as the preservation of the overall scalability of the new network solutions, the definition of the control protocols for routing and distributing information on the identified traffic flows and finally, the main concern in this work, the definitions of flow and service identification mechanisms.[8] One of the basic requirements for the future Internet routers is the ability to pinpoint applications that require different service levels. The ability to detect and classify individual traffic flows from the network, preferably at the application level, might result in added value to these applications, provided we could also offer differentiated service levels.

Several studies have been conducted on different traffic classification methods.[9–13] These studies mostly concentrate on optimizing the performance of the routing component in the Internet router. The studies either explicitly or implicitly presume that traffic flows which contain a large amount of packets are beneficial to the user if prioritized

Other author information: (Send correspondence to M.I.)
M.I.: E-mail: lynx@tct.hut.fi
Fax: +358 9 451 2474

and, to this end, the classification schemes aim to detect flows of this kind. Very little emphasis is put to study the actual effect that these traffic classification methods have to the user of the network and, to be more specific, what are the applications that are classified. Our main intention in this work is to study the effect that a particular traffic classification scheme has both to the network elements (IP router) and to the user.

In this work, we first define the concepts and the analysis methodology used in this work. We then develop the heuristics that we use in the multi-class classification and briefly observe the properties and suitability of the Learning Vector Quantization (LVQ) algorithm to the traffic classification problem. Using our heuristics we work our way to develop the multi-class Internet traffic classifier using the LVQ algorithm. Finally, we study the quantitative and qualitative performance of the multi-class LVQ traffic classifier.

2. KEY CONCEPTS AND PERFORMANCE ANALYSIS METHOD

2.1. IP application

Since the applications in the Internet are diversified in their performance requirements, it is necessary to pinpoint the applications needing prioritized handling with adequate accuracy. Some of the connection oriented and connectionless Internet applications that use the TCP-protocol and the UDP-protocol as the transmission protocol utilize the set of Well–Known TCP/UDP port numbers. Furthermore, many of the recent applications utilize the rest of the available port space quite freely. All this means that detecting the applications using the TCP/UDP port numbers for better service is somewhat problematic. However, the current network protocols do not offer any other means of detecting individual applications from the network without explicit user intervention.

The IP application, in this work, is defined by the TCP/UDP source port number. This definition of the IP application makes the concept server–oriented. If the definition of the application would also take into account the destination port number we would experience an abundance of identical applications since the destination port numbers are used in various ways depending on the implementation of the receiving TCP/UDP client.

As a final note here, we point out that the choice of TCP/UDP (source) port as the application identifier is arbitrary and is used to enable us to get simulation results from real traffic traces. In the future, one could also use IPv6 flow labels or some other means to ensure that the sending application is distinctly identified in the packet headers.

2.2. Traffic classification

We define traffic classification to be a process of observing various statistics in the network and extracting information out of these observations with the intention of using this information to guide the packet and flow classification processes and set rules for these processes.

Traffic classification could be considered as a process that remembers the past and based on this information on the past tries to resolve (near) future network traffic properties with the aid of various analysis methods for the data from the past (measurements). In practice, traffic classification process determines the criteria, or the rules, with which either the flow or the packet classification idnetifies a packet with a particular prioritization policy.

Several methods of traffic classification have been previously introduced. Typical to the work done so far is that it has mostly concentrated on dividing the traffic into two parts; the default, best effort traffic, and the prioritized traffic. These classifiers are shortly summarized in Table 1. The end result of the traffic classification process, seen from the user point of view, is the network service profile (NSP).

2.3. Network Service Profile

The Network Service Profile (NSP) in its simplest form is a list or a set of IP application identifiers determined by the traffic classification process, and used by the flow or packet classification. The NSP might also contain information about the policy with which the individual packets of particular IP applications are dealt with. The separate mechanisms of resource allocation and connection admission control (CAC) then determine the criteria with which individual flows are treated. To achieve practical functional networks the allocation of resources is detrimental to the network. To allocate resources appropriately requires knowledge of the source behavior. This knowledge can be obtained from the user or it might be determined beforehand by analyzing the characteristics of the source. This interesting and important issue is beyond the scope of this work.

Table 1. Internet Traffic classifiers

Classifier type and nature	**Triggering event**	**Other notes**
Network resource usage optimization -oriented		
Packetcount,[9–11,14] static	Packetcount threshold per individual flow.	Per-flow based.
Dynamic Packetcount,[13] dynamic	Packetcount threshold per individual flow.	Per-flow based, threshold modified by the use of Software forwarding, Connection set–up and Active connection resources.
User satisfaction / NSP -oriented		
List classifier,[15] static	First packet matching to NSP criteria	NSPs are selected from an arbitrary, but static, portion of applications having the largest packets to flows -ratios. May be applied in the aggregated traffic environment.
Static service profile,[9–11,16] static	First packet matching to NSP criteria	NSP formed manually by network administration. May be applied in the aggregated traffic environment.
2-class LVQ,[16,15,17] dynamic	First packet matching to NSP criteria	Changing NSPs determined with LVQ algorithm. May be applied in the aggregated traffic environment.

Ideally we would want to obtain and use an NSP containing a set of applications that benefit the user if the traffic of these services would be classified: Specifically, the user benefits if the classification method is able to detect and classify traffic according to its characteristics to the appropriate class.

In real networks, the functionalities of the packet and traffic classification lie in the physical path the bits traverse when going through a router. The use of the traffic classification system in the path of the packet in an Internet router is shown in Figure 1. Typically, the determination of the NSP would be done off-line, whereas the actual NSP would be used in conjunction with the flow and packet classification in real-time.

The NSP may be updated either manually by the network manager or automatically by some method that monitors the network and learns the applications used, and their characteristics, in the network. Depending on the case, we may be talking about static or dynamic NSPs respectively. The automatic construction of the NSPs could basically be done in two different ways:

1. The process might be implicit; the NSP would be obtained as a side result of performing some other related task in the network: For example, the NSP could be formed while working towards forwarding workload reduction using a packet count flow classifier.[9–11,18,14] The prioritized traffic flows would then form the NSP. In these methods, it is usually assumed that it is beneficial to the user if traffic flows with large number of packets are prioritized. However, despite the fact that the network equipment is functioning in an optimal fashion, the user may still not be experiencing any significant performance improvement.

2. The process of defining the contents of the NSP might also be explicit: we would then be using a specific method to form the NSP. The method might be based on network managers personal insight to the network and its service profile, or the NSP could be constructed based on information obtained and processed from traffic measurements. A measurement based traffic classifier gathers statistical data from the network to adapt to the traffic properties of the network it is connected to. In measurement based traffic classification the NSP is constructed automatically by observing the network traffic and using specific analysis methods.[15]

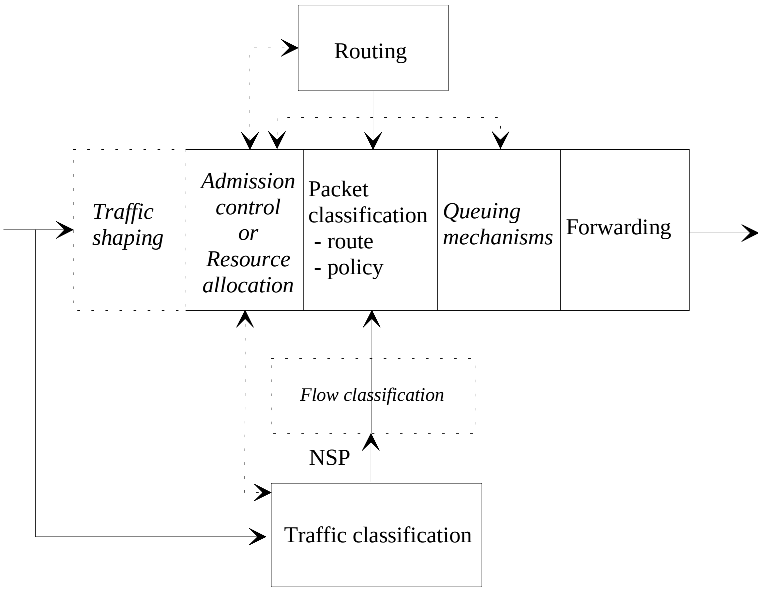

Figure 1. Key concepts of traffic classification in an Internet router

2.4. Performance analysis

From the router performance side, the traffic classification scheme must preferably be able to detect services containing the majority of packets seen in the network so the router workload may be reduced as much as possible via caching the similar decisions concerning these service flows. At the same time, we should not be moving the workload from the routing and forwarding to the mechanisms dealing with the classified packets. In the connectionless environment we should take care that the amount of classified packets does not rise too high effectively reducing the benefits of classification. In the connection-oriented environment we need to monitor the connection setup resources as opposed to the use of packet forwarding resources.

To adequately observe the quantitative performance of a traffic classification scheme we use a quadrilateral model containing general factors of quantitative performance.[15] The measured factors enable us to observe and compare different classification schemes in relation to each other. The following four factors are determined:

- *Connection setup factor* gives an indication how much the average connection setup speed and the number of connections needed to set up in the trace differ from the reference point where all flows seen in the network would be given their own connections.
- *Connection space factor* shows what is the maximum number of simultaneous connections in the network for the observed traffic.

  The connection setup factor and the connection space factor relate closely to the connection-oriented technology. These two factors lose their relevance if the network environment does not offer dedicated connections to the classified packets and therefore needs no separate connection space.
- *Prioritization factor* indicates how many packets in the total traffic are prioritized in a particular service class. This factor indicates the overall efficiency of the traffic classification scheme from the point of view of forwarding workload reduction in a connection oriented IP router environment. The more packets are mapped to less connections means that the forwarding component and route look-up procedure of the IP router are experiencing a lesser demand for performance. However, in a connectionless environment for the prioritization to be successful, a traffic classification scheme should not prioritize too many packets. Otherwise, the benefits of packet prioritization are lost and no user service differentiation is achieved.
- *Application factor* indicates, according to the IP application definition earlier, how many different applications are present in the trace and how they are mapped to a particular prioritization or service class. This factor

gives an indication on how the user is experiencing the traffic classification scheme in the quantitative sense. An abundance of applications might very likely result in an ambiguous NSP and might be wasting connection space and other resources.

To visualize the four quantitative factors we use the quadrilateral model for comparing different traffic classification schemes in Figure 2. The quadrilateral model gives us an illustrative picture on how the different traffic classification schemes perform in relation to each other. The ends of axis present the maximum of that factor in a certain network (simulation) environment.

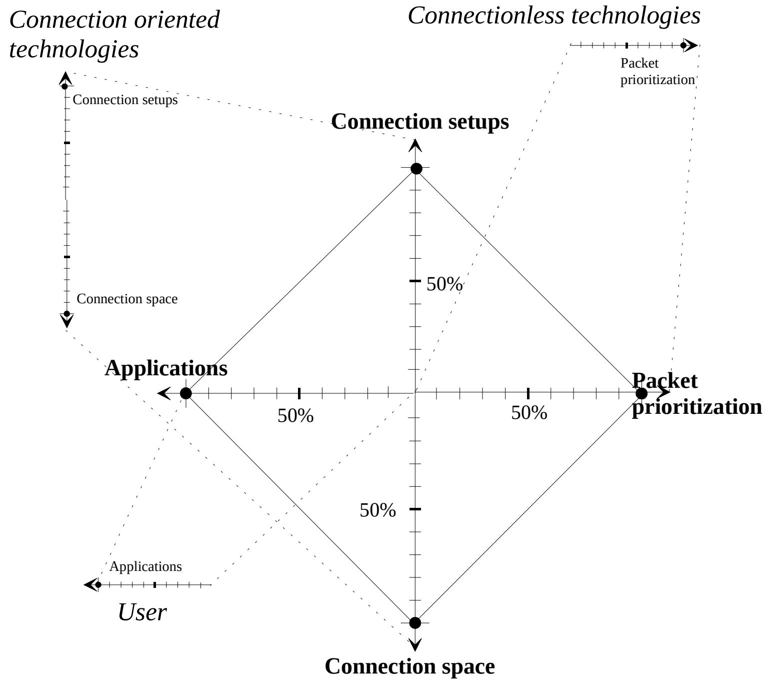

Figure 2. Quantitative performance model for comparing traffic classification methods

From the users' point of view, the traffic classification process should also be able to offer the users of the network perceptible increase in the observed or perceived performance of the applications creating the classified traffic. Otherwise, we could just as well concentrate on optimizing the performance of the network components without any added value to the user. Therefore, it is important to make sure that if traffic classification is used in the network, it will also provide network users and applications an increased service level. The first step to realize this would be to implement methods to differentiate traffic groups from the traffic. The quality of the classified applications would then be evaluated by inspecting the NSPs that the traffic classification scheme has determined.

3. MULTI-CLASS LVQ CLASSIFICATION

3.1. The principles of flow analysis for multi-class classification

The process used for constructing the classifiers and classifying the traffic into multiple classes starts by first analyzing, for each application, the number of flows and the packets on these flows respectively with different flow timeout values. The number of flows (per application) contains information on the usage of this particular application in the network, but since the number of flows is depending on the flow parameters, especially the flow timeout value, it also contains information on the temporal behavior of the application. Therefore, in this work, the flow timeout value is varied in order to differentiate traffic.

The shorter the timeout value is the fewer number of TCP connections or packets in a UDP stream can be mapped under one flow, in fact a flow might, in some cases, contain only parts of TCP connections / UDP streams. Used in this way the flow could be used to detect applications that send or need to send packets with very little interval between them. If we classify flows that send packets at an even rate and that do not pause for a lengthy periods of time we may consider this kind of application as extremely interactive (hard-interactive). To use a relatively low flow timeout value we might be able to detect these traffic flows from the network as having a relatively large number of packets mapped into a relatively small number of flows and thus we might be able to detect flows of interactive applications. However, if the application sends packets with longer intervals in between packets or if the related traffic flows are detected wide apart from each other, we could argue that the applications are not interactive by nature and do not need a high level of prioritization. The longer the timeout of the flow is, the more TCP connections or packets of UDP streams can be aggregated in a single flow.

Following these heuristics, *we claim and base our forthcoming work on the assumption that applications with large numbers of packets and low count of flows with a particular flow timeout value possess interactive qualities more than the services with fewer packets and larger number of flows with the same timeout value.* This statement is clarified in Figure 3.

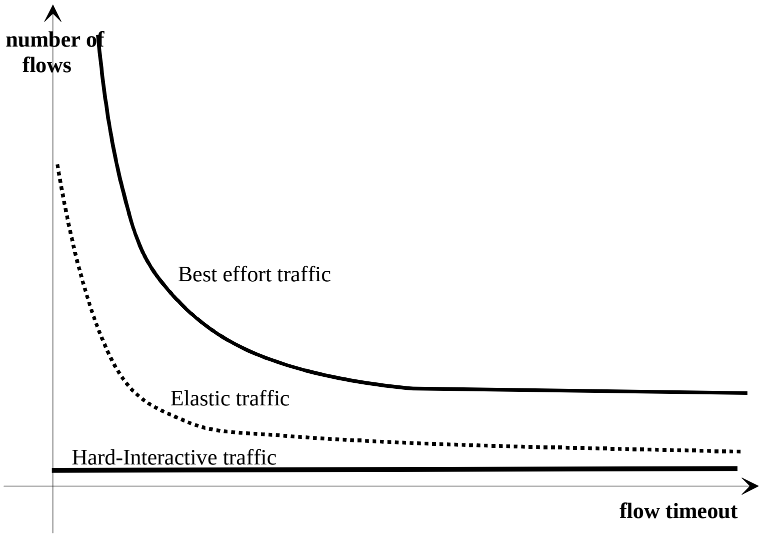

Figure 3. The effect of flow timeout to the number of observed flows

For each application identifier, we measure the number of flows and packets with flow timeout values of 0.1s, 1s and 10s. This division and the values of flow timeouts are arbitrary and might be altered depending on the network where the traffic classification would be applied. The application identifier (TCP/UDP source port), the number of flows and the number of packets form the vectors to which we apply the LVQ algorithm. To achieve generality of the results we normalize the results to the total number of packets and flows respectively. Finally, we present these network traffic profiles with the measurement results to the classification process (multi-class LVQ classifier) which will return the NSP for each service or prioritization class.

3.2. Characteristics of Learning Vector Quantization

The k–nearest–neighbor ($k$nn) methods for classification have provided good performance on a variety of real-life data sets and often perform better than more complicated approaches. Two possible reasons are stated in [19]:

1. Practical problems often have a low intrinsic dimensionality although they may have many input variables. If some of the input variables are interdependent, the data lie on lower-dimensional manifold within the input space and this effectively reduces the dimensionality of the problem.

2. Accurate estimates of conditional probabilities are not necessary for accurate classification. In some sense, the problem of classification is not as difficult as regression, so the effect of dimensionality is less severe.

Table 2. Traffic traces analyzed in this work

Trace information				
Name	*Location*	*Length*	*Nr of packets*	*Other information*
ebb900	Helsinki University of Technology (HUT)/Campus Area Network (CAN) 09.00 May 29, 1997	1 hr	1.1 million	Ethernet
dec-pkt-1 (dec1)	Digital's primary Internet access point (DIAP) 22:00, March 8th, 1995	1 hr	2.9 million	Ethernet; WWW–archive[a]

[a] All dec-pkt-x traces are freely available at http://ita.ee.lbl.gov/html/contrib/DEC-PKT.html

For problems with many data samples, classifying a particular input vector using ($k$nn) poses a large computational load, since it requires storing and comparing all the samples. In traffic classification we aim to classify several thousands of applications and doing this application by application would consume a lot of effort. One way to reduce the load is to represent the large dataset by a smaller number of prototype vectors. This approach requires a procedure for choosing these prototype vectors so that they provide high classification accuracy. The solution provided by Kohonen in [20] is to use learning vector quantization (LVQ) methods to determine initial locations of prototype vectors, then assign class labels to these prototypes, and then adjust the locations using a heuristic strategy that tends to reduce the empirical misclassification risk.[19] The purpose of the LVQ algorithm is to define class regions in the input data space and is therefore meant to be a method for statistical classification or recognition. To this end, a subset of similarly labeled codebook vectors is placed into each class region.

An example of a traffic classifier using the LVQ algorithm is the 2-class LVQ classifier.[16,17,15] The 2-class LVQ classifier updates the prioritized service set according to measurements made in the network thus being able to adapt to changes in the network traffic.

In the LVQ traffic classifier no tight feedback loop exists; the experience of the user on the applications classified is not readily and immediately available and depends on various factors, and no other connection between the packets to flows -characteristics and the NSP exists. Performance observation is as easy as with the other classification schemes. However, with the LVQ classifier a substantial interest lies also in the quality of the NSP.

3.3. Building the multi-class LVQ traffic classifier

The traces used in the simulations are described in Table 2. As we apply the LVQ algorithm to the traffic classification problem we use the basic program package* explained in [21]. For a detailed description of the application of the LVQ algorithm to traffic classification for two classes the reader can refer to [15].

The teaching sets are constructed manually by observing the complete application list and choosing clear candidates for default handling and prioritized handling. The teaching vectors for the two traces used in this work are shown in Figure 4. We want to emphasize here that we only choose two classes, the default and prioritized, of teaching vectors and by consecutive classification with the 2-class LVQ traffic classifier, we obtain the final three classes. This method has been chosen bearing in mind that if the LVQ traffic classifier would be used in a real network environment it would be far more easier for the network manager to pick clear examples of prioritized traffic than to complicate the process by trying to prioritize the already prioritized examples into several classes.

The LVQ classifier is then applied in the network to perform multi-class classification as illustrated in Figure 5 where the progression of the classifier construction and application process is shown.

The construction begins with the traffic measurements and flow analysis on the measurements. From the analysis results we pick up clear candidates for prioritization and form the teaching vectors (refer also to Figure 4). With the aid of the teaching vectors, we use the LVQ algorithm to form the 2-class traffic classifier that is then applied to the results of flow analysis with the 0.1s, 1s and 10s flow timeouts. The traffic classification process advances

*Available via WWW from http://www.cis.hut.fi/nnrc/lvq_pak/

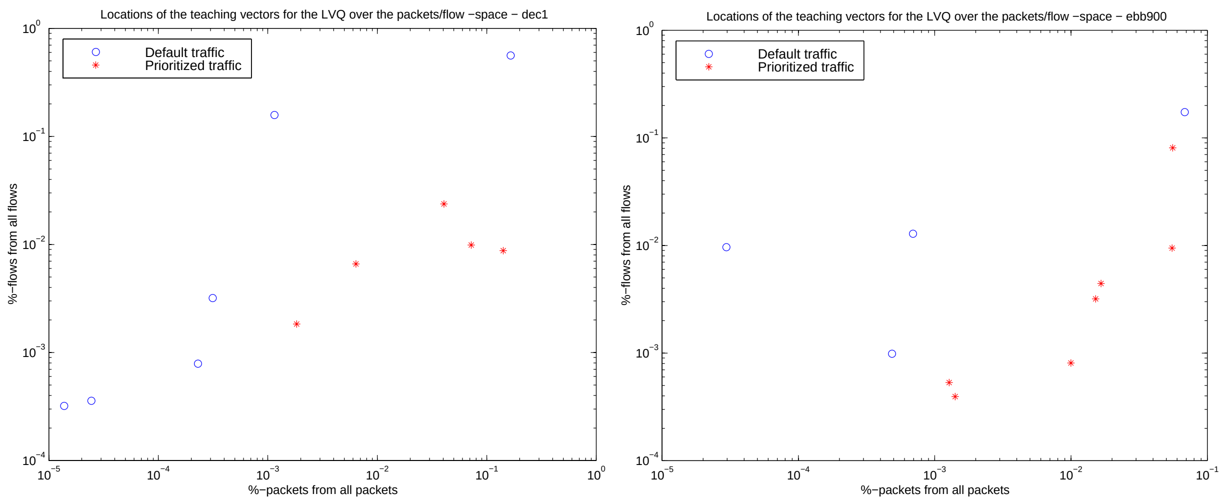

Figure 4. Location of the teaching vectors for the multi-class LVQ-classifier / dec1 and ebb900

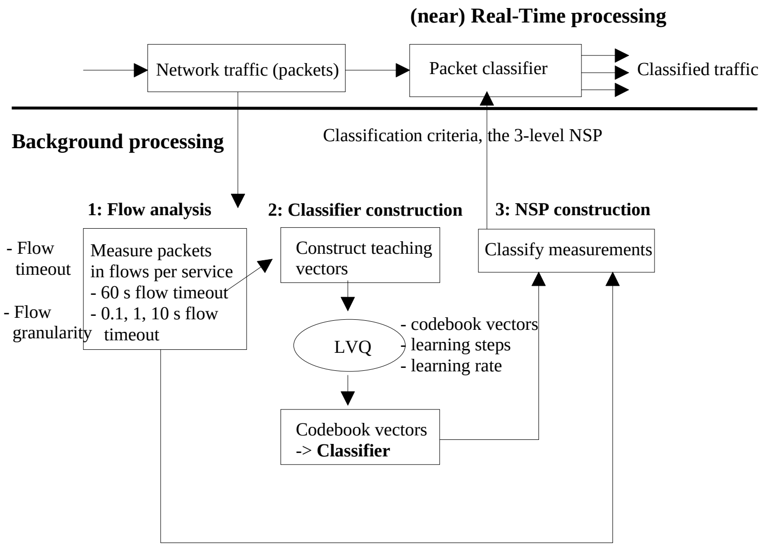
(near) Real-Time processing
Network traffic (packets)
Packet classifier
Classified traffic
Background processing
Classification criteria, the 3-level NSP
1: Flow analysis
2: Classifier construction
3: NSP construction
- Flow timeout
- Flow granularity
Measure packets in flows per service
- 60 s flow timeout
- 0.1, 1, 10 s flow timeout
Construct teaching vectors
Classify measurements
LVQ
- codebook vectors
- learning steps
- learning rate
Codebook vectors -> Classifier

Figure 5. Framework for applying the multi-class LVQ classifier in the network

in combining the classification decisions on each timeout value to form a classification string. By using majority decisions, we use the classification results to divide the traffic into three classes that are shown in Table 3. The applications that get the prioritized decision in every level of flow timeout get to be classified to the first class. Then all of the services that get the default classification decision get to be classified to the third class. All other traffic, which contains one prioritization decision, is classified to the second class. Essentially, with each timeout value, we are looking for applications that have a large packets to flows -ratio and claim that these services are important to the user.

The final classification results give us the two prioritized classes and the third, best-effort class, is formed of those applications that did not get into either of the prioritized classes.

Table 3. Service classes used in the 3-class LVQ traffic classifier

Service class properties in the multi-class LVQ classifier		
Class nr	*Intended Class properties*	*Classification string*[a]
1	Hard-Interactive, applications that send packets at very short intervals for lengthy periods of time.	$p \wedge p \wedge p$
2	Elastic, applications that part of the time send packets at very short intervals, but have moderate intervals between bursts.	$\neg((p \wedge p \wedge p) \vee (d \wedge d \wedge d))$
3	Best Effort, the rest of the applications	$d \wedge d \wedge d$

[a] p stands for prioritization and d for default traffic

Table 4. Basic quantitative performance analysis for the multi-class LVQ classifier

Performance statistics for the multi-class LVQ classifier				
Classified network	*Flow factor*	*Connection space*	*Prioritization factor*	*Service factor*
dec1 / class 1	9,55%	11,73%	41,69%	0,64%
dec1 / class 2	16,73%	13,04%	4,44%	0,57%
ebb900 / class 1	14,26%	19,26%	69,71%	0,57%
ebb900 / class 2	18,32%	19,40%	14,01%	0,49%

4. PERFORMANCE ANALYSIS

4.1. Quantitative analysis

In Table 4 the performance statistics for the two prioritized classes in the multi-class LVQ classifier for the two studied networks are shown. We can see that the packet prioritization factor is quite high in the first class. The amount of services detected is low and the use of connection space and connection setups is reasonable.

Furthermore, we can see that the number of services prioritized in the traces changes very little between the networks. It can also be seen that the average number of packets in the first class is higher than in the second class and it is not unusual to have over 50% of the packets assigned to the first class.

In Figure 6, we observe the per-class performance statistics of the multi-class LVQ classifier in the quadrilateral model. We see that in the first class for the dec1-trace the classifier picks out the applications containing the majority of packets. These applications seem to be long-lasting since the use of the connection setup resources is quite low. For the first class in the ebb900-trace, the packet prioritization factor is very high and we can also observe a relatively high usage of connection setup resources.

When observing the performance of the second class classification process we see a significant change to the first class behavior. The use of connection setup resources compared to the amount of packets prioritized is higher and the packet prioritization level is significantly lower than in the first class.

The size of the NSP (classified applications) seems to be very small in both of the prioritized classes. This indicates that the classification process has been successful in detecting those applications that generate the majority of the packets seen in the traces.

4.2. Qualitative analysis

In Table 5 we observe the complete per-class NSPs for each network. Looking at Table 5 we can see that there are some consistent candidates for prioritization but that most of the applications, especially in the higher port-number regions, change from trace to trace. This reflects to a conclusion that in the detailed level the same applications

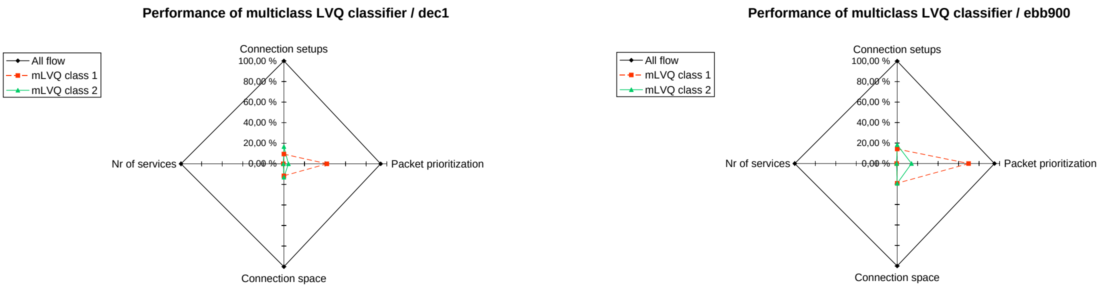

Figure 6. Performance of the multi-class LVQ classifier for the class 1 and class 2

Table 5. Per-class NSPs obtained with multi-class LVQ classifier / dec-traces

Complete service lists for multi-class LVQ classifier / dec-networks	
Classified network	*NSP*
dec1 / class 1	20, 21, 23, 25, 80, 119, 520, 1042, 1126, 1144, 1307, 1397, 1599, 1628, 1979, 2128, 2326, 2547, 2754, 2770, 2845, 3012, 3083, 3084, 3143, 3737, 3980, 4128, 4281, 4324, 4436, 4521, 4923, 4968, 4971, 8080, 41381
dec1 / class 2	123, 513, 514, 540, 1038, 1047, 1048, 1049, 1057, 1058, 1064, 1065, 1093, 1163, 1202, 1296, 1352, 2017, 2183, 2451, 2460, 2692, 3001, 3549, 3655, 3868, 4184, 4382, 4495, 4517, 41170, 41304, 42911
ebb900 / class 1	0, 20, 52, 79, 80, 82, 119, 801, 1008, 1015, 1070, 1216, 1424, 1560, 1928, 1929, 1935, 2143, 2156, 2188, 2198, 2213, 2220, 3281, 3403, 3789, 3961, 3969, 3971, 4580, 4581, 4588, 4592, 6000, 44232
ebb900 / class 2	22, 23, 53, 1019, 1020, 1021, 1023, 1026, 1028, 1030, 1031, 1037, 1056, 1218, 1235, 1258, 1565, 1821, 1938, 2159, 3074, 3967, 4183, 4184, 4185, 4700, 4866, 11150, 15017, 19157

are used varyingly in different networks at different times. This would implicate that the NSP is local phenomena and its use should be restricted to an autonomous area of the network. Furthermore, it might be useful to detect continuous areas in the port-number space that get to be prioritized.

In terminal access protocols the shift from telnet -protocol (port 23), which is classified to the first class in the *dec1* -trace and to the second class in the *ebb900* -traces is explained partly because between the traces the traffic has distributed to the recently introduced ssh -protocol in the ebb -traces and the diminishing amount of terminal traffic in general reflects the movement towards the usage of the http-protocol. The ftp-data service (port 20) acts consistently throughout all of the networks getting first class classification in all of the traces.

As a clear example of the potential in the multi-class LVQ classifier one can observe the detection of the alternate HTTP -port (8080) in the dec-trace and the X-win -application (port 6000) in the ebb-trace, two clearly interactive application protocols, to the first class.

Finally, we observe in Figure 7 how the prioritized applications map onto different classes in the flows/packets –space. The data points shown are taken from the flow analysis with a timeout value of 60 seconds. However, the original classifications were based on flow analysis done on flow timeout values of 0.1, 1 and 10 seconds. This explains in part the overlapping of the classified services that is observed in Figure 7. In general, we can see that the services having a relatively high packet count against a relatively low flow count are classified higher than services with opposite characteristics.

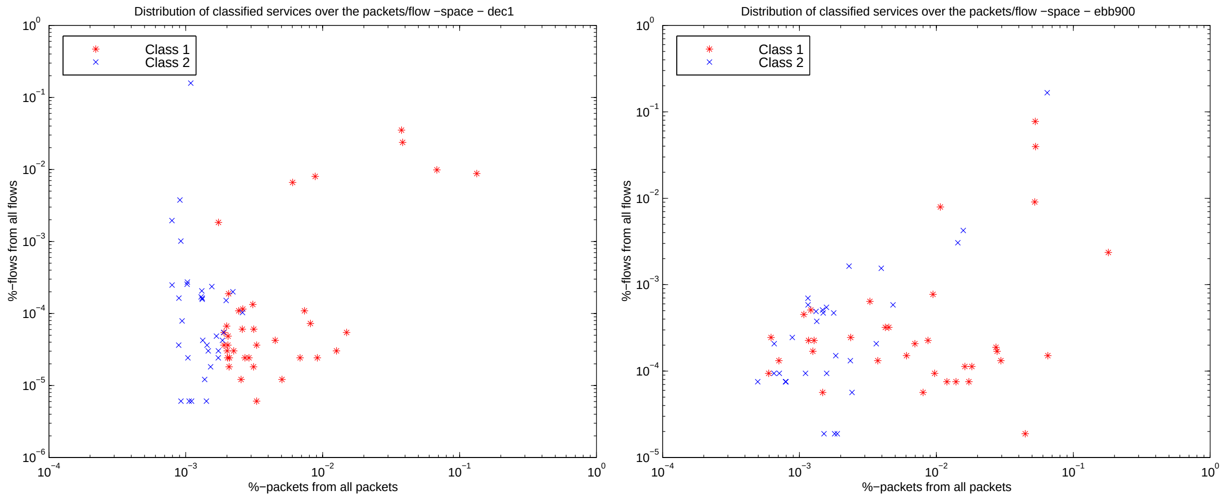

Figure 7. Location of services using the multi-class LVQ-classifier / dec1 and ebb900

5. CONCLUSIONS

Starting from the suspected behavior of interactive applications, we have developed a method to apply 2-class LVQ classification repeatedly to measurements done with differing flow timeout values. This method has shown promising results as it has succeeded to classify applications that the user of the network might hold important and that could be stated to be of interactive nature. However, the traffic traces used are somewhat outdated and lack any truly interactive applications, such as Voice over IP. In addition, the teaching samples used are determined from the existing applications and the results presented here must be viewed against these restrictions.

The multi-class LVQ classifier is able to pick out applications outside of the teaching set and with the proper and educated choosing of the teaching vectors the LVQ classifier can be fine-tuned to pick out applications with desired behavior in the packets/flow –space.

The results show that a large portion of the packets are to be prioritized. For the network to be able to provide CoS or QoS, however, the amount of packets needs to be strictly restricted. Furthermore, because the method used in this work (measuring the number of packets and flows per service) guides the creation of NSPs, but the use of NSPs does not directly guide the total packets flows statistics or restrict the use of applications, it might be argued that using a traffic classification scheme like this directs the usage of applications to the kind that produce large quantities of packets in few number of flows. Therefore, the results indicate that additional service architectures and related business models with proper incentives to regulate the traffic and packets to the prioritized traffic classes are necessary. On the other hand, the optimization of forwarding component in the IP router would be very successful; a lot of packets can be mapped to a relatively low count of connections (flows).

Finally, we emphasize that the results for the multi-class classifier contain an unrealistic aspect. As the NSP is determined only after the traffic analysis is performed, and only after this, the results are simulated for performance analysis we obtain unrealistic results. In real networks, we could not state the NSP beforehand. In actual deployment the results would be less satisfying than has been illustrated here.

On a network architecture level we believe that in order to provide end-to-end QoS, in addition to accurate traffic classification, a connection oriented link layer technology is needed. These technologies are able to offer dedicated forwarding paths for individual packets thus reducing the processor intensive hop count and flattening the network.

Future research in using the multi-class classification should advance. Research issues include the extension of the measurement data dimensionality, automated teaching vector creation, and applications of LVQ classification in different network service architectures.

ACKNOWLEDGMENTS

Mika Ilvesmäki is supported by the Nokia Foundation and by Technology Research Center from the contract for project IPANA. Marko Luoma is supported by Finnish Academy from the contract for project MI^{2}TTA.

REFERENCES

1. K. Kilkki, "An introduction into the philosophy of differentiated services." www-nrc.nokia.com/sima/, March 1998.
2. W. Weiss, "QoS with differentiated services," *Bell Labs Technical Journal - Packet Networking* **3**, pp. 48–62, October-December 1998.
3. P. P. White, "Rsvp and integrated services in the internet: A tutorial," *IEEE Communications Magazine* **35**, pp. 100–106, May 1997.
4. P. P. White and J. Crowcroft, "The integrated services in the internet: State of the art," tech. rep., University College London, 1996.
5. T. Lakshman and D. Stiliadis, "High–speed policy–based packet forwarding using efficient multi–dimensional range matching," in *Proceedings of SIGCOMM '98*, IEEE/ACM, IEEE/ACM, 1998.
6. H. H.-Y. Tzeng and T. Przygienda, "On fast address-lookup algorithms," *IEEE Journal on Selected Areas in Communications* **17**, pp. 1067–1082, June 1999.
7. K. ichi Nagami, H. Esaki, Y. Katsube, and O. Nakamura, "Flow aggregated, traffic driven label mapping in label-switching networks," *IEEE Journal on Selected Areas in Communications* **17**, pp. 1170–1177, June 1999.
8. E. Guarene, P. Fasano, and V. Vercellone, "IP and ATM integration perspectives," *IEEE Communications Magazine* , January 1998.
9. P. Newman, T. Lyon, and G. Minshall, "Flow labelled ip: A connectionless approach to atm," in *IEEE Infocom, San Francisco*, IEEE, March 1996.
10. S. Lin and N. McKeown, "A simulation study of IP switching," in *ACM SIGCOMM '97*, 1997.
11. M. Ilvesmäki, K. Kilkki, and M. Luoma, "Packets or ports - the decisions of IP switching," in *Broadband Networking Technologies*, S. Civanlar and I. Widjaja, eds., vol. 3233, pp. 53–64, SPIE, SPIE, November 1997.
12. J. Karvo and M. Ilvesmäki, "Nondeterministic classifier performance evaluation for flow based IP switching," in *High Performance Networking*, H. R. V. As, ed., pp. 613–624, Kluwer Academic Publishers, 1998. IFIP TC-6 Eighth International Conference on High Performance Networking (HPN'98) Vienna, Austria, September 21-25,1998.
13. H. Che, S.-Q. Li, and A. Lin, "Adaptive resource management for flow–based IP/ATM hybrid switching systems," *IEEE/ACM Transactions on Networking* **6**, pp. 544–557, October 1998.
14. P. Newman, G. Minshall, and T. L. Lyon, "IP switching – ATM under IP," *IEEE/ACM Transactions on Networking* **6**, pp. 117–129, April 1998.
15. M. Ilvesmäki, R. Kantola, and M. Luoma, "Adaptive flow classification in IP switching: The measurement based approach," in *Internet Routing and Quality of Service*, R. O. Onvural, ed., vol. 3529 of *Proceedings of SPIE*, SPIE, SPIE, November 1998.
16. M. Ilvesmäki, M. Luoma, and R. Kantola, "Flow classification schemes in traffic–based multilayer IP switching — comparison between conventional and neural approach," *Computer Communications* **21**, pp. 1184–1194, September 1998.
17. M. Ilvesmäki, M. Luoma, and R. Kantola, "Learning vector quantization in flow classification of IP switched networks," in *IEEE 1998 Global Telecommunications Conference*, vol. 5, pp. 3017–3022, IEEE, IEEE, November 1998.
18. H. Che, S.-Q. Li, and A. Lin, "Adaptive resource management for IP/ATM hybrid switching systems," in *Broadband Networking Technologies*, S. Civanlar and I. Widjaja, eds., vol. 3233, pp. 328–339, SPIE, SPIE, November 1997.
19. V. Cherkassky and F. Mulier, *Learning from Data - Concepts, Theory and Methods*, The Wiley Series on Adaptive and Learning Systems for Signal Processing, Communications and Control, John Wiley & Sons, 1998.
20. T. Kohonen, *Self-Organizing-Maps*, Springer Series in Information Sciences, Springer–Verlag, 2 ed., 1997.
21. T. Kohonen, J. Hynninen, J. Kangas, J. Laaksonen, and K. Torkkola, "LVQ_PAK: The learning vector quantization program package," Tech. Rep. Report A30, Helsinki University of Technology, 1996.