BANDWIDTH OPTIMIZATION BY TRAFFIC CLASSIFICATION INMULTIMEDIA BROADBAND NETWORKS

N.Akhtar andM.Kamran

University of Engineering and Technology Lahore, Lahore 54890, Punjab, Pakistan
Corresponding author e-mail: akhtar.naveed@uet.edu.pk

ABSTRACT: In last decade, high bandwidth consumption of Peer to Peer (p2p) applications has resulted increased bandwidth demandand service quality issues for critical business applications. Past researchsuggestedthat traffic filtering of unwanted traffic and rate limitation of bandwidth hungry applications can result in fair distribution and optimization of available bandwidth. In this context we proposed and implemented internet traffic rate limiter and bandwidth optimizer based on FreeBSD (v8.1) operating system. The prioritized traffic flows throughput was increased by 5% and network latencywas reduced by9.5%, which improved quality of service of prioritized traffic flows.

Key words*:* Internet bandwidth optimization, Network security, Internet traffic prioritization, Deep packet inspection, Network performance evaluation.

(Received 17-08-2015 Accepted 11-07-2016)

INTRODUCTION

Network security and effective utilization of available bandwidth always remained a big challenge when multimedia traffic is mixed over Internet Protocol (IP) networks. The solution to the problem is to classify network traffic and to allocate available bandwidth as per defined traffic flow priority (Nguyen and Armitage, 2008). There are different methods for bandwidth optimization; some of them are traffic shaping (Luo *et al*., 2008), traffic flow prioritization(Williams *et al*., 2006), caching and traffic rate controlling(Li and Moore, 2007). Bandwidth control by port number is sometimes difficult because applications can be designed to use random port numbers to communicate and notto be recognized by traffic shapers (Stewart *et al*., 2005). In order to overcome this problem each packet of the data is inspected (Divakaran*et al*., 2015) and researchers have started working on statistical properties (Dainotti*et al.*, 2008) of different flowsfor accurate classification of internet traffic.A detailed survey work for identification of internet traffic by using statistical characteristics of different traffic flows has been done by Nguyen and Armitage, (2008). Li and Moore, (2007) have suggested machine learning approach based on Naive Bayes and C4.5 decision tree algorithms, which accurately classify the internet traffic by collecting different features at the start of internet traffic flow. Nguyen *et al*,. (2012) used machine learning for dynamic identification of different traffic flows using their statistical characteristics. Zander *et al*.,(2005) worked on traffic classification based on machine learning. Sen*et al*.,(2004) worked on identification of P2Ptraffic using application level signatures and designed online filters that were able to track P2P traffic with accuracy and robustness. Jiang and Gokhale,(2010) implemented Locality Sensitive Hashing (LSH) on Field Programmable Graphics Arrays (FPGAs) and suggested that computational complexity of internet traffic classification using statistical approaches based on machine learning can be high and Internet Service Providers (ISPs) may not deploy them in their systems. Moore and Zuev, (2005) used Bayesian analysis technique for Internet traffic classification and achieved 90% accuracy on training data of different applications.

The identification of encrypted traffic is biggest challenge to the traditional port based classification techniques (Bernaille*et al.*,2006) and for this reason the classification techniques based on machine learning (Wright *et al.*,2004) and statistical properties of different flows are preferred.Dainotti*et al*., (2008) used statistical properties, i.e. inter packet time and packet size of different traffic flows for traffic classification using Hidden Markov Model (HMM).The key advantage of HMM is the ability to analyze more than one parameters of a state. For modeling of these random parameters,HMM "λ" is describedas set of states (S), Transition Matrix (A), Vector of initial state probabilities (π) and Emission model (B). The StateTransition Matrix, $A=\{a_{ij}\}$, where A is $N*N$ matrix (Stamp *et al*., 2004) and N is the number of states in the model, Where a_{ij}= P (state$a_j$at t+1| state $a_i$at t), the emission model B is described by the observation Probability matrix= $\{b_j(k)\}$, which is $N*M$ matrix, where N is the number of states and M is the number of observation symbols with $b_j(k)$, which is given as under

$b_j(k)$ = P (observation k at t | state $q_j$at t), whereas π_{x0} (π representing initial state distribution) is theprobability ofinitialstate x_0. The general form of Hidden Markov Model is shown in Figure-1., where $O_i$is representing the observations related to the different

states of the Markov process. The different features of FreeBSD have been covered by Carbone and Rizzo, (2010) in his work.

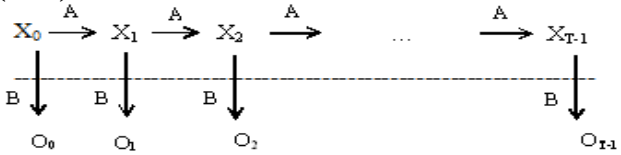

Fig-1: Hidden Markov Model

MATERIALS AND METHODS

The internet traffic flows passed through three stages of packet processing as shown in Figure-2. At each stage the processes were invoked for identification, shaping, filtering and policing of different flows.

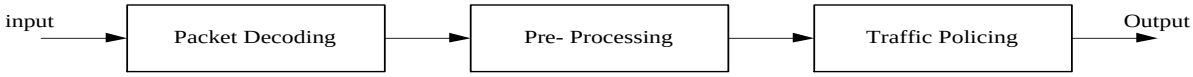

Fig-2: Packet processing stages

The proposed framework identifies various traffic flows and prioritized these flows as per configured priority. In FreeBSD v8.1, Ipfw (IP firewall) user interface was used for controlling the Packets, which entered the firewall from different places in the protocol stack. The traffic flowing through the firewall was compared against all the rules in the rule set according to the rule-number, order permitted and in the order of insertion of packet as shown in Figure-3. The packets were inspected and matched against different configured rules, and if the match was found, then packet was treated according to the matching rule.In order to rate limit internet traffic, traffic passedfrom two objects i.e. pipe and a queue. The objects configured for emulation purpose were pipe; the pipes emulate the link with certain delay and bandwidth, and then traffic further passed to the scheduler.

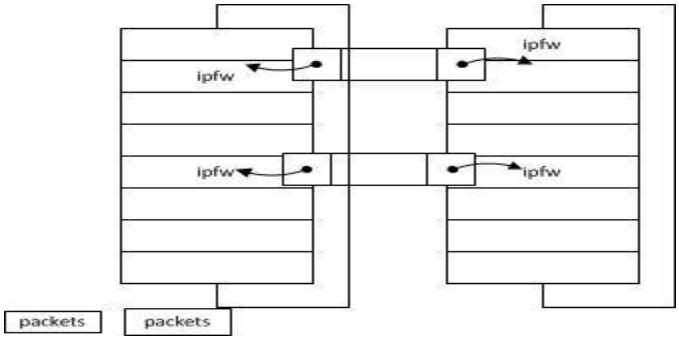

Fig-3: Packets flowing through different stacks.

The key variables for a pipe are queue size, link bandwidth and the network end to end delay. Each pipe was assigned a numeric identifier as its identity and had finite queuing capacity as per available memory. The packet selection was made using ipfw program. The traffic was originated from different machines and captured at FreeBSD server shown in Figure-4.

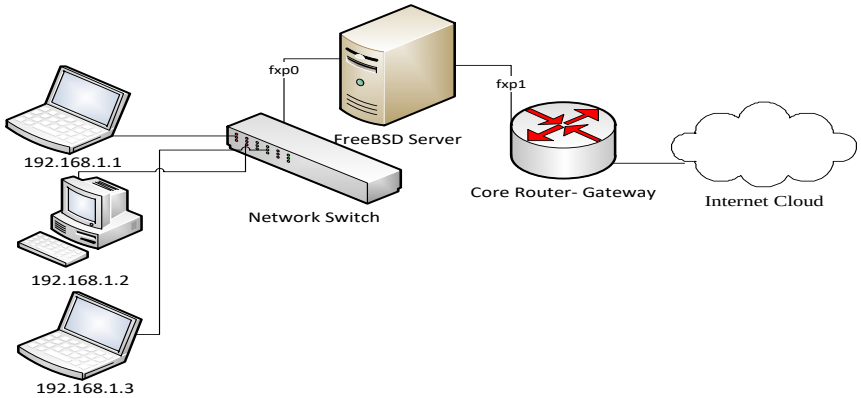

Fig-4: Network Diagram

The FreeBSD server has two network interfaces; fxp0 and fxp1. The fxp0 was connecting inter network devices, whereas fxp1 was connected to the public internet through gateway router.

Packet Classifier: The traffic passed to the pipes using classifier, ipfw, which used configured ruleset to match the incoming packets and to decide the required actions(accept, pass or drop). All configured options were evaluated sequentially on incoming traffic and traffic flows were dealt as per configured rules. Once the ruleset was configured, the ipfw route traffic towards the specific pipe; later on packets were injected back into the network stack, the rule numbers were configured as below

ipfw add rule-number actions options

In below ruleset, the rule numbers 110 and 220 were passing bidirectional traffic (through pipe no 10 & 11) to the host xyz.it, these rules were configured in FreeBSD kernel

ipfw add 110 pipe 10 out dst-ip xyz.it
ipfw add 220 pipe 11 in src-ip xyz.it

Pipe Configuration and Rate limiting: The pipe configurable parameters were bandwidth, delay and queue size as is shown in Figure-5.

For traffic rate limitation of client 192.168.1.1, Pipe 10 was configured for outgoing traffic and Pipe 11 for incoming traffic as is shown in Figure-6.

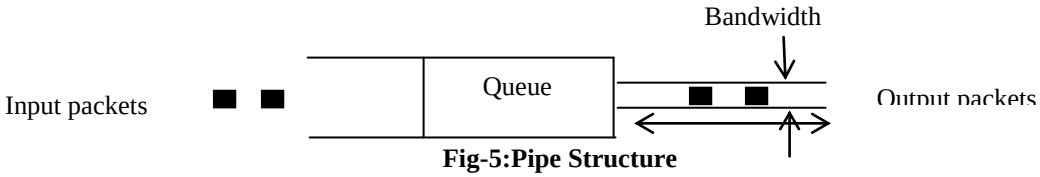

Fig-5:Pipe Structure

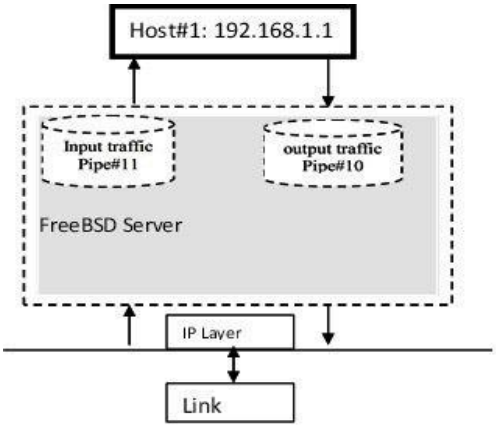

Fig-6:Client 192.168.1.1 Input and output traffic pipes

The configuration steps were as under

 # cd /etc
 # vi ipfw.ruleset
ipfw add 110 pipe 10 out 192.168.1.1
ipfw add 220 pipe 11 in 192.168.1.1
 # Client 192.168.1.1 was restricted to output rate of 4.5Mbps and input rate of 0.5Mbps
ipfw pipe 10 configbw 4500Kbit/s delay 5ms
ipfw pipe 11 configbw 512Kbit/s delay 15ms
 # save and exit file
:wq!

For client 192.168.1.1, output Pipe "10" was configured for 4.5Mbps output rate, which means that the client 192.168.1.1 was rate limited to 4.5Mbps and when the overall output rate of all clients exceeded the overall available bandwidth at output interface fxp0 then queuing took place and the excess traffic was queued and it was transmitted later on. The average queue time calculated using Little's Law (Robertazzi, 2000) is as under

$$N = \lambda T \quad (1)$$

Where λ is average packet arrival time, N represents the average number of packets in queue and T is average delay in queue. Moreover the approximate number of packets in queue at any time can be found as per equation (2)

$$E = [N(t)] = \sum_{n=0}^{\infty} n.P\{N(t)\} = \sum_{n=0}^{\infty} n\ p_n\ (t) \quad (2)$$

Traffic Flow Prioritization: To ensure service quality for critical applications, test traffic for TCP and UDP was prioritized using Wf2+q queuing technique and following

rules were configured in "ipfw.rules" for priority implementation. The priorities were assigned to queue "7" and queue "8" by assigning different weightages to these queues.

```
# vi ipfw.ruleset
ipfw add 00015 sched 10 config type wf2q+
ipfw queue 7 config weight 20 sched 10
ipfw queue 8 config weight 10 sched 10
ipfw add 00016 queue 7 out proto udp
ipfw add 00019 queue 8 out proto tcp
# save and exit file
:wq!
```

RESULTS AND DISCUSSION

Traditional IP networks does not provide service quality guarantee offer;in this studythe service quality improvementwasbandwidth guarantee for business critical customers. The required link capacity wouldde pend on different flows packet scheduling and queue management at core routers. Traditionally, FIFO (First in First out) was used for scheduling and RED (Random early detection) was used for packet dropping and to control the overall traffic flow.WFQ (Weighted Fair Queuing) scheduler was used to prioritize different flows to provide service guaranty. WFQ provided the link capacity proportional to the assigned weight against each traffic flow.To describe FIFO and WFQ results further, a set of connections flows (w_i) that flow through a link "m" of capacity $c_l^{(FIFO)}$ would get the share of the link capacity given as $r_i^l = (\frac{w_i}{\sum_{j \varepsilon F_l} w_j})(c_l^{FIFO})$, whereas$w_i = (\frac{1}{(\tau_i \sqrt{h_i}}$) was the weight assigned to each traffic flow against total allocated bandwidth, $\tau_i$is representing RTD (Round trip Delay) and h_i was representing the number of congested links. The results for link capacity utilization and how did all applications compete for available bandwidth were discussed by Floyd, (1991) and Mathis *et al*., (1997).For traffic prioritization, traffic was offered to available queues configured as WFQ and results were measured for packets round trip time (RTT), the Timestamp sent and its echo reply time moving average was used to measure RTT.

Fig-7: Timestamp for TCP traffic

To calculate throughput of available bandwidth under normal queuing it was assumed that TCP's congestion window adaptive mechanism changed the window size, and in case of packet loss, it reduced to one

half of its window size "*w*" with a constant packet loss rate probability "*p*". Under these assumptions different flows congestion windows behaved as a periodic sawtoothas is shown in Figure 8.In a study conducted by Nelson, (2013) illustrated similar results by considering different flows as independent random stochastic processes.

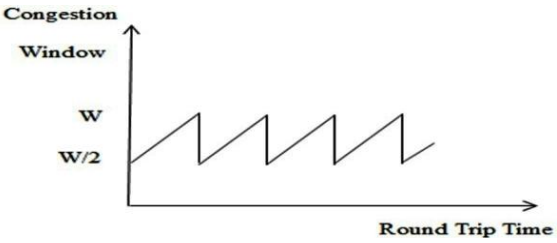

Fig-8: Behaviorof TCP congestion Window

The number of TCP connections throughput ("r") is given as under

$$\frac{No\ of\ Packets\ per\ Cycle}{Time\ per\ Cycle}=\frac{Area\ under\ Sawtoot\ h\ waverform}{Time\ per\ cycle}$$

$$r = \frac{(\frac{3}{8}W^2)}{\tau\frac{W}{2}}=\frac{0.867}{\tau\sqrt{p}}$$ packets per seconds. (3)

The observed throughput pattern of each flow wasin linewith the work conducted by Benmohamed*et al.,* (1998)through implementation and follow gamma distribution as is shown in Figure-9 and different traffic throughput was calculated using equation (3) mentioned above.

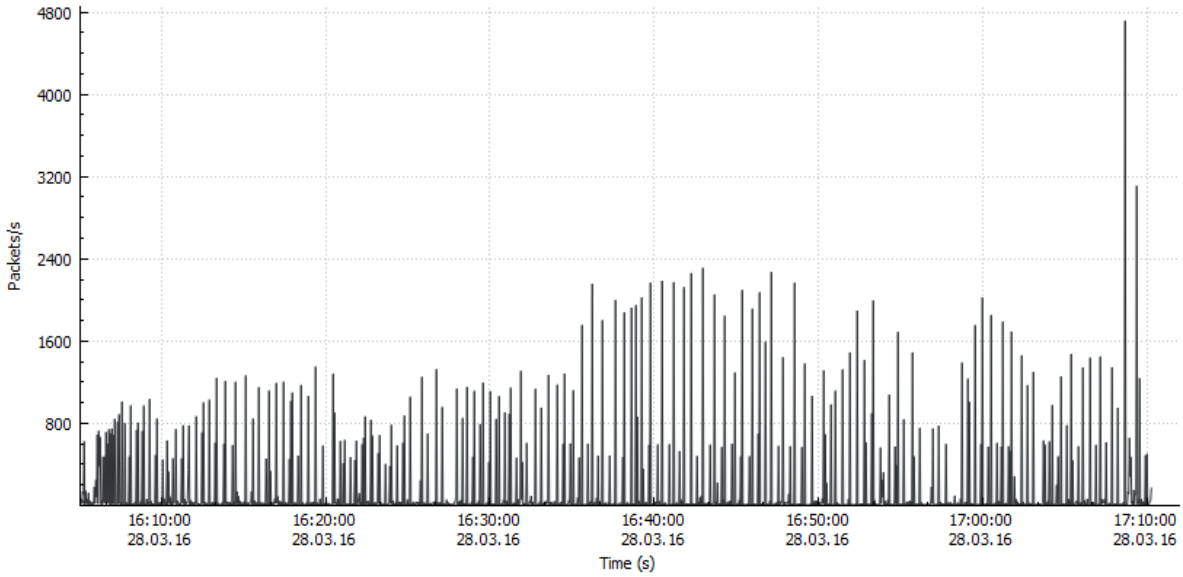

Fig-9: Traffic pattern of different traffic flows.

To calculate different flows network latency before and after prioritization using WFQ, test traffic was generated and it was observed that overall network latency of prioritized traffic flows was reduced; thenetwork latency increased as the packet size was increased asis shown in Figure-10. The overall network latency before and after applying ruleset is also shown in Table 2.

Table-2: Network Latency before and after WFQ implementation

Packet Length (Bytes)	No of Packets	Percentage of total traffic	Network Latency(ms)	Rate (ms)-After applying ruleset	Overall Impact on Network Latency
0-79	3079	44%	0.000912	0.000897	1.645%
80-319	1898	27%	0.000152	0.000131	13.816%
320-639	719	10%	0.000168	0.000151	10.119%
640-1279	298	4%	0.000191	0.000173	9.424%
1280-2559	1019	15%	0.000274	0.000239	12.774%

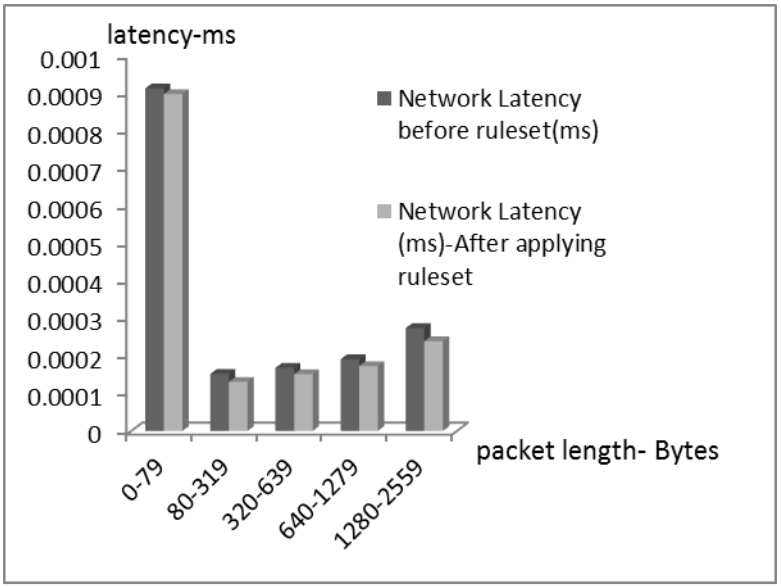

Fig-10:WFQ impact on Network latency

As shown in Figure-10., the WFQ implementation for different traffic flows resulted 5% improvement in overall available bandwidth utilization by using queues. Similar results were worked out by Carobneand Rizzo, (2010)by dummynet implementationfor different queue management policies and presented how dummynetwould perform under different conditions. The 5-tuple packet properties, i.e. source IP, source port, destination IP, destination port and the protocol in use were used to control different traffic flows. Nguyen and Armitage(2012) calculated similar results for traffic identification using statistical characteristics of different traffic flows which wereconsistent with observed results.Divakaran*et al*., (2015) used weighted k-NN (Nearest Neighbor) in their proposed SLIC (Self-Learning Intelligent Classifier) and 80-85% accuracy was achieved for the identification of different traffic flows. The results achieved were in line with observed results.

Conclusion: In this study, Network bandwidth optimization was implementedby applying traffic rate controlling on different internet applications using FreeBSD (v8.1) operating system. Though deep packet inspection was among the available solutions but for Internet Service Providers; solution complexity, cost, network load (utilization) and legal requirements were major constraints towards DPI (deep packet inspection) implementation. The Quality of Service was improved by prioritizing businesscritical applications using WFQ and 9.5% overall improvement in network latency was achieved.This research work would help further to explore FreeBSDapplicationsinother areas of internet traffic classifications like network security, usage based billing and network planning.

REFERENCES

Benmohamed, L.M., S. Dravida, P.Harshavardhana, W.C.Lau and A.K.Mittal (1998).Designing IP networks with performance guarantees. Bell Labs Technical Journal, 3(4):273-296

Bernaille, R. Teixeira and K. Salamatian(2006).Early application identification. Proceedings of the ACM CoNEXT conference, ACM

Carbone, M. and L.Rizzo (2010).Dummynet revisited. ACM SIGCOMM Computer Communication Review. 40(2):12-20

Dainotti, A., W. Donato, A. Pescape and P. S. Rossi (2008).Classification of network traffic via packet-level hidden Markov models. Proceedings of Global Telecommunications Conference, IEEE GLOBECOM: 1-5

Divakaran, D. M., L.Su, Y. S. Liauand V.L.Thing (2015). SLIC: Self-Learning Intelligent Classifier for Network Traffic. Computer Networks, 91,283-297

Floyd, S.(1991). Connections with multiple congested gateways in packet-switched networks part 1: One-way traffic. ACM SIGCOMM Computer Communication Review, 21(5):30-47

Jiang, W. and M. Gokhale(2010).Real-time classification of multimedia traffic using fpga. International Conference on Field Programmable Logic and Applications, IEEE: 56-63

Li, W.and A. W. Moore (2007).A machine learning approach for efficient traffic classification. Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, MASCOTS'07. 15th International Symposium. IEEE: 310-317

Luo, Y., K. Xiang and S. Li(2008). Acceleration of decision tree searching for IP traffic classification. Proceedings of the 4th ACM/IEEE Symposium on Architectures for Networking and Communications Systems, ACM: 40-49

Moore, A.W. and D. Zuev (2005).Internet traffic classification using bayesian analysis techniques.ACM SIGMETRICS Performance Evaluation Review, ACM, 33(1):50-60

Mathis, M., J. Semke.J. Mahdavi and T. Ott(1997). The macroscopic behavior of the TCP congestion avoidance algorithm. ACM SIGCOMM Computer Communication Review,27(3):67-82

Nguyen, T. T., G. Armitage, P. Branch and S. Zander(2012).Timely and continuous machine-learning-based classification for interactive IP traffic. IEEE/ACM Transactions on Networking (TON), 20(6): 1880-1894

Nelson, R. (2013). Probability, stochastic processes, and queueing theory.the mathematics of computer performance modeling. Springer Science & Business Media

Nguyen, T. T. andG. Armitage (2008).A survey of techniques for internet traffic classification using machine learning. Communications Surveys and Tutorials, IEEE,10(4): 56-76

Robertazzi, T.G. (2000). Computer networks and systems: queuing theory and performance evaluation. Springer Science & Business Media

Stewart, L., G. Armitage, P.Branch and S. Zander(2005).An Architecture for Automated Network Control of QoS over Consumer Broadband Links. IEEE Region 10 Conference (Tencon):1–6

Stamp, M. (2004). A revealing introduction to hidden Markov models. Department of Computer Science San Jose State University,USA.

Sen, S., O. Spatscheckand D. Wang (2004).Accurate, scalable in-network identification of p2p traffic using application signatures. Proceedings of the 13th international conference on World Wide Web,ACM, 512-521

Williams, N., S. Zander and G. Armitage (2006).A Preliminary Performance Comparison of Five Machine Learning Algorithms for Practical IP Traffic Flow Classification. SIGCOMM Computer Communication Review, 5-16

Wright, C., F. Monrose and G. Masson (2004). Hmm profiles for network traffic classification, in VizSEC/DMSEC. 9–15

Zander, S., T.Nguyen and G. Armitage (2005). Automated traffic classification and application identification using machine learning. IEEE Conference on Local Computer Networks, 30th Anniversary, 250-257.