# Automatic VLAN creation based on on-line measurement

Sean Rooney Christian Hörtnagl Jens Krause

IBM Zurich Research Laboratory
Säumerstrasse 4
CH-8803 Rüschlikon/Switzerland
{sro,hoe,jkr}@zurich.ibm.com

**Abstract**

Virtual LANs (VLANs) permit hosts connected to a LAN switch to be grouped together into logical groups as a function of some management policy rather than simply of their physical location. Commercial LAN switches support a variety of policies based on either physical or logical addresses, protocol types, tagged frames, or user defined rules. The objective of these policies is the same: to reduce the amount of traffic that needs to be routed by grouping together hosts which are likely to communicate with each other into the same virtual LAN. This paper proposes a novel and more direct approach, it shows how VLANs can be created and removed dynamically as a function of the *measured* traffic patterns across the network. This is both simpler than configuring many static rules and permits the VLAN configuration to adapt to the evolution in the traffic patterns. The latter point is especially important in future LANs supporting peer-to-peer continuous media services, such as IP telephony or video-conferencing, in which clusters of hosts come together to communicate with each other intensively for relatively short periods of time and then form into new clusters.

## 1 Introduction

In a single shared Ethernet segment, the offered load (defined as the average number of hosts waiting to transmit) increases as a function of the number of transmitting hosts.

Traffic patterns often involve clusters of hosts or applications, with frequent communication between members, but sparser communication to the outside. Consider this corporate network as an example: Typical hosts participate in client/server protocols, such as NFS or HTTP. If big files are involved, it is straightforward to define a cluster, with outside communication concerning traffic such as e-mail (or again HTTP). When another subset of hosts is used for video-conferencing, their amount of mutual communication may dominate and suggest forming a second cluster. The point equally applies to other protocols as well.

The example suggests that there are in effect two separate networks, which happen to share the same broadcast domain. The behavior of one network interferes with the other because of the presence of broadcast traffic (used by protocols such as ARP) and aggregate network load on shared media.

Switched Ethernet addresses these problems by allowing a LAN to be partitioned into associated local segments. Each local segment attached to the switch behaves like a normal Ethernet LAN, but the switch determines which frames should be forwarded between local segments.

It is constraining if the behavior of a host must be tightly coupled with its location on the network, for example a company might wish to place all its servers in the same secured room, making it problematic to attach them to different LAN segments although for efficiency reasons this might be the best configuration.

Ethernet switches can learn about which MAC addresses are reachable through which ports and make intelligent forwarding decisions. However, any sufficiently large switched network still needs to be partitioned into broadcast domains, because although the amount of overall traffic which is broadcast may be relatively small, for a large enough network the amount of useless traffic a given host receives will have a detrimental effect on its performance. For example, an IP network is divided into subnets, such that each subnet is one broadcast domain; traffic between subnets is routed.

Modern Ethernet switches [1] support the concept of a virtual LAN (VLAN), — first introduced in [2] — whereby hosts which are physically separated can be associated into logical groups, so for example a set of diskless clients can be placed on the same VLAN as their server, without requiring that they all be attached to the same physical segment.

The switch determines which VLANs an attached host should be associated using a set of rules, normally defined by the network manager. In this paper we propose a mechanism of using these rules and combining them with traffic measurements and analysis to arrive at an efficient assignment of hosts to VLANs, relative to shifting traffic loads. Its benefit in terms of speed improvements depends on the relative cost of taking forwarding decisions at level 2 or level 3. Related advantages of switching over routing have recently been challenged by router implementations that achieve fast layer 3 forwarding with ASICs and optimized address lookup [3]. These silicon-enhanced implementation offer better performance but also incur higher complexity and cost [4] as well as inflexibility when protocols evolve. We observe nevertheless that the distinction between switching and routing has diffused and become

somewhat less significant overall. Another important contribution to this development stems from work on IP Switching [5], Tag Switching [6], and the currently active Multiprotocol Label Switching (MPLS) standardization effort. Our approach is similar to these cut-through techniques on the outset, but focuses on a particular context, i.e. the current generation of switched Ethernet (Section 3 elaborates more on this contrast). Shifting the balance between routed and switched traffic on Ethernet was one part of our motivation for this work. A second resulted from a larger effort on VPN creation in heterogeneous networks. VCs are proper substrates for achieving resource reservation on ATM. Future enhancements to Ethernet will include suitable provisions as well (see Section 6.2). In the meantime VLANs give us a way to cover deployed switched Ethernet within the same framework, and therefore extend its reach from WANs to LANs.

Commercial Ethernet switches support many ways to define which VLAN(s) a host should be associated. For example the IBM 8277 Ethernet RouteSwitch include rules based on:

- the port to which the host is attached;

  the MAC address of the host;

  the IP address of the host;

  the high-layer protocols the host is using;

  user defined rules, based on bit patterns within the frames

A given host can at any one moment belong to multiple VLANs. Other vendors [7] offer a similar and (equally proprietary) portfolio of rules. VLANs are currently being standardized [8] so that LAN switches from different vendors can interoperate.

The rule-based method of determining the VLANs with which a host is associated has the advantage of being easy to implement on the switch. For example, if a rule has been defined that all hosts on a particular IP subnet should be associated with a given VLAN, then when the switch receives a frame with an unknown source MAC address, it can check to see if the destination IP address in the frame belongs to the same subnet and if so add the port through which the MAC address is visible to the IP VLAN. Henceforth all broadcast traffic from that port will be forwarded to only those ports on the same VLAN.

However, these rule based systems have the following disadvantages:

there must be some piece of information easily deduced from an arriving frame in order to determine to which VLAN the emitting host should belong. Not all useful host groups can be characterized using only the information in a normal Ethernet frame ([8] has addressed this problem by proposing the inclusion within the frame header of a VLAN tag).

in order to characterize a range of useful VLANs, rules can be defined at many different levels e.g. from the MAC layer to the user layer, many different rules can be in force on the switch simultaneously, making it complicated for a network manager to determine a priori to which set of VLANs a host will belong.

they are not very adaptive. For example if a rule is in place that all members of the IP subnet should be associated with a given VLAN, then in order to remove a given host from that VLAN and place it in another it is necessary to changes the hosts IP address.

although hosts can be dynamically assigned to VLANs, the division of the LAN into VLANs is carried out statically when the network operator assigns the rules.

All the problems listed above are symptoms of the same cause: the intention is to group together hosts which intercommunicate the most, using static rules which at best only give indications of the likely behavior of the hosts. Such a system is entirely adequate when traffic patterns are easily defined a priori and are reasonably static, e.g. when most traffic on the LAN is between workstations and NFS, HTTP etc. servers. However it has severe drawbacks in a system in which there is much peer-to-peer communication and when it is difficult to determine in advance which peers are likely to wish to communicate; future LANs offering continuous media service such as video-conferencing are likely to have these properties.

This paper proposes determining which hosts are likely to communicate together in the future (and in consequence in which VLANs they should belong) by constantly measuring with which hosts they are *currently* communicating. So for example, if two hosts that are located in different subnets start to communicate intensively with one another, then the system can dynamically create a VLAN containing the two hosts and in consequence further traffic between them will be switched rather than routed. The policy as to when the system should make a 'cut through', e.g. how frequently the updates should take place, how much traffic between how large a set of hosts must be routed before the cut through should take place, is decided by the network operator.

As the observation is carried out at the Ethernet layer no notion of a 'flow' is available for the process, instead simply the bulk amount of traffic carried between the various hosts in the network is measured and used. The work takes as an axiom that if a set of hosts have been communicating a lot in the 'recent' past, then it is probable that they will continue to do in the 'near' future. This is simply assumed to be intuitively true and no mathematical analysis is presented, however recent work [9] has shown that it is possible to predict future traffic patterns from current ones.

The practicality of this approach is motivated by describing an initial implementation of such a system over our test network. A comparison with similar techniques in Section 3 demonstrates our approaches' uniqueness; some initial results are also given.

## 2 System Overview

This section briefly outlines the general concept of automatic VLAN creation using on-line measurement. As many of the issues cannot be easily discussed without reference to an implementation, they will be reserved for Section 4.

Figure 1 shows the relevant sequence of interactions required to automatically generate a set of VLANs using on-line measurement.

Associated with each switch in the network is a switch controller running on or local to the switch. The switch controller periodically (with a period defined by the network operator) gathers statistics about the information being currently carried across the switch (1); the basic information required is a matrix containing the amount of communication which takes place between each known host.

One of the switch controllers is designated or elected as the master switch controller. All switch controllers send their statistics to this master (2), which then uses this information in order to partition the hosts into groups.

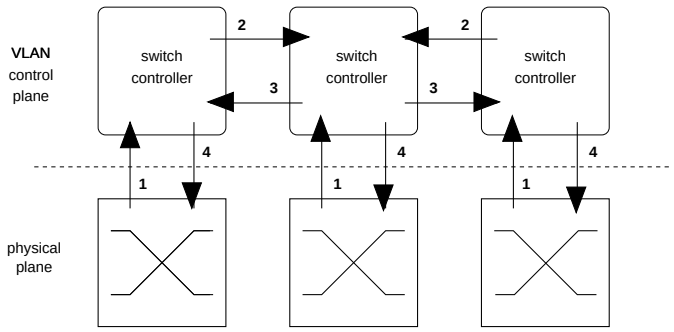



Figure 1: Overview of Architecture

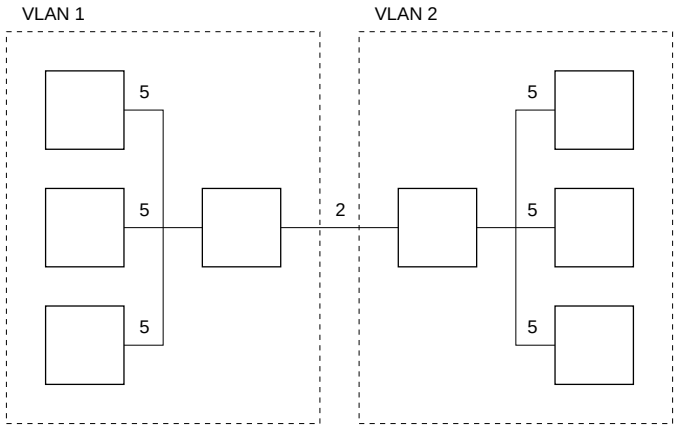



Figure 2: Example of a Partition. Numbers indicate units of communication between hosts.

The goal of the partitioning is to keep the size of the group at or close to some ideal number of hosts while at the same time ensuring that the inter-group communication does not exceed some upper bound. The ideal number of hosts and the upper bound are configuration parameters of the system. It may not be possible for a given set of statistics to satisfy these constraints, in which case they are weakened (by increasing the ideal number of hosts and increasing the communication upper-bound) and the algorithm is re-applied.

The problem is NP-complete — it is equivalent to the Minimum K-Capacitated Tree Partition in [10] — as such no method can guarantee an optimal solution for an arbitrary configuration. Section 2.1 gives a general description of the algorithm used in the current implementation, it should be noted that this is only by way of motivation and no claim is made that this is the best approach.

When the master switch controller has determined an acceptable partition of the hosts into groups, it defines a VLAN for each group, uses this definition to create the VLAN on its own switch and passes the definition to other switch controllers (3) so that they can update themselves (4).

### 2.1 Partitioning Algorithm

Figure 2 shows how 8 hosts would be partitioned into VLANs with ideal size of 4 and with maximum inter-VLAN communication of 2 units. In the configuration shown 15 units of communication are transmitted within each VLAN and 2 units must be routed between VLANs. Note that if the maximum communication had been set to 1 or the ideal group size had been set to 2 then the constraints could not have been met.

The partitioning algorithm used is a form of greedy algorithm; it starts by placing all the hosts into singleton groups, it creates a sequence of pairs of groups ordered as a function of the number of octets they exchanged within the observed period. Pairs are merged if the size constraint is not transgressed. The process is then repeated with the merged groups until it is no longer possible to merge any groups further.

As initial experience suggested that only comparing two nodes in isolation often led to solutions far from the one that a human would recognize as optimal. For example, the greatest single amount of communication might be between two servers (say if one is the mirror of the other), while the *total* amount of communication that their clients communicates with each of them far outexceeds that. The greedy algorithm would start by placing the two servers in the same VLAN and the resulting solution would not be the obvious one.

Heuristics were introduced in order to reduce this problem. These heuristics try to identify clusters of hosts by comparing the amount of communication a group of hosts G has with its immediate neighbors N and the amount of communication that GG, formed by merging G with N, has in turn with its neighbors NN.

The set of G's to apply the operation above to are formed by placing a host in a singleton group and successively adding the neighbors of the group to form new G's. This way be applied starting from all hosts, some small number (e.g. the the first three of the hosts ordered by the number of neighbors they have) or just one.

These cluster were then used as the starting groups for the greedy algorithm.

## 3 Comparison with other Techniques

This Section briefly outlines other techniques based on the use of creating a cut through and compares them with the current proposal.

### 3.1 IP Switching

IP Switching [5] allows an IP router running over an ATM switch to optimize long lived IP flows by mapping them onto ATM connections. The packets are switched in the ATM switch rather in the IP router, thereby increasing efficiency. The technique proposed in this paper is similar in that it seeks to map strongly communicating groups onto a VLAN, within which traffic can be switched rather than routed. However both the switching technology (ATM or Ethernet) and the context (WAN or LAN) changes the problem significantly.

An ATM virtual channel at a given switch is realized by associating a identifier — the Virtual Channel Identifier (VCI) — with each ATM cell in the channel, this tells the ATM switch to where the cells in the virtual channel should be forwarded and the resources that should be allocated to the cells in the virtual channel in its pas-

sage from the input to the output port across the fabric. The identifier has local significance only; as part of its forwarding function the switch swaps the actual identifier in the cell for one of significance to the next switch downstream, i.e. it changes the input port VCI of the cell to the output port VCI. An end-to-end ATM virtual circuit is a set of such virtual channels.

In some ways an Ethernet VLAN resembles an ATM virtual circuit; the Ethernet identifier may be implicit (determined from other information in the Ethernet frame or contained IP packet) or explicit as specified in [8] but it either case it helps the switch in determining to which output ports the frame should be forwarded. Moreover, in [8] the identifier (or 'tag' as it is termed) allows a decision to be made about not only where VLAN frames should be forwarded but also what resources should be allocated to them (i.e. in which priority forwarding queue they should be placed).

However, the Ethernet VLAN identifier (either implicit or explicit), has more than a local significance; all Ethernet switches supporting a given VLAN must agree on its identifier. So in contrast to IP Switching, the current proposal examines parts (or potentially the entire) managed network in order to determine how it should be partitioned. While this is clearly impossible for the WAN environment (that for which IP Switching is proposed) it is possible within a LAN environment as it involves gathering information from a comparatively small number of switches all resident within a single trusted community.

So while an IP Switching enabled router attempts to distinguish individual IP flow from amongst the totality of routed traffic and mapping them onto ATM virtual channels, the current proposal involves periodically reconfiguring the entire LAN so that the VLANs better reflect the traffic patterns observed in the previous period. As the entire LAN is considered the definition of 'communicating enough to be on the same VLAN' can be comparative rather than absolute.

While the computational complexity of the VLAN partitioning is to a great extent dependent on the algorithm used, clearly the current proposal is more complex and computational intensive than IP Switching. However, as there is at least a magnitude of difference in the timescales on which they are required to function this is not a significant drawback; whereas IP Switching has to make its decision as to whether to create a cut through virtual channel on very short timescales and bases that decision on only the actual or recently observed flows, the current proposal might only reconfigure every hour, day or week. The length of the period during which the traffic should be observed, the periodicity with which the network should be configured and the algorithm for determining the partition are all independent of the technique itself.

Section 5 gives some results related to the performance of the implemented algorithm in relation to its ability to detect and partition clusters of tightly communicating groups within an Ethernet network as a function of network operator specified parameters.

In summary, IP Switching and Automatic VLAN Creation share common aims: decreasing the amount of routed traffic, and common means: observing the network traffic patterns then modifying the switch state in order to create cut through. However, the constraints imposed by the their distinct environments are such, that they are very different in realization.

### 3.2 MPOA and NHRP

In ATM LAN Emulation [11] two hosts connected to the same switch may be on different Emulated LANs and therefore on different subnets. All communication between the hosts must therefore be routed. MPOA [12] allows virtual circuits to be created directly between hosts in different ELANs. It makes use of the Next Hop Resolution Protocol (NHRP) [13] to obtain information, in particular the ATM address, about entities still within the same physical ATM network but outside its subnet. NHRP may be thought of as a 'better ARP'.

While in ATM it is sufficient to obtain the end address of a remote host to create a cut through using signaling, as Section 3.1 described, no analogous mechanisms exists for Ethernet network.

## 4 Implementation

Section 2 outlined in very general terms the architecture of the system; this section will motivate the practicality of such a system by describing an actual implementation using commercial Ethernet switches.

The proof-of-concept implementation uses IBM 8277 Ethernet RouteSwitches; such switches have a fairly typical set of capabilities for the support of VLANs. PCs running Linux kernel 2.0.35 are used as the hosts. In addition, a NetScout 6030 RMON [14] probe is used for statistic gathering. In the network all the PCs are directly attached to the Ethernet switch through a full duplex 100 Mb/s port, i.e. all hosts have a dedicated LAN segment.

The network is manually configured such that all hosts and switches are part of a single IP subnetwork, which corresponds to the 8277's default VLAN. This IP network is termed the *public network* in the rest of this paper and a host's IP address within this network is termed its *public address.* As the 8277 is a commercial Ethernet switch, it is not possible for third parties to enhance its control software, therefore in the proof-of-concept implementation the switch controllers were run as separate processes on attached workstations. Communication between controller and switch took place using SNMP.

Each switch controller needs to discover the overall topology of the physical network in order to carry out its task. The IP addresses of all transmitting hosts in the system can be obtained from the 8277 internal book keeping records, which maintain a set of IP/MAC/switch port triples. Each switch controller sends out a topology request message to all of these IP addresses requesting whether the connected device is running a switch controller and if so what is the switch controllers current view of the network. A switch controller updates its view of part of the network using information from another switch controller, if the latter has a less ambiguous view of that part of the network. After obtaining a complete view of the network each switch controller decides if it should be the master switch controller. It does this by comparing the number of neighbors its switch has with those of other switch controllers; the switch controller whose switch has the greatest number of neighbors is the master switch controller. If two switch controllers have the same number of neighbors, then a tie breaker rule is used (in the current implementation the switch with the highest IP address wins).

Although the switch itself offers some basic RMON functions, it does not support the RMON *Matrix* group which identifies who is communicating with whom and how many octets they are sending. Therefore an external NetScout RMON probe was used by the switch controller to gather statistics about the traffic patterns across the public network. It is to be expected that future Ethernet switches will offer more extensive RMON support. As this probe is only capable of monitoring a single port at a time, the switch controller "samples" the traffic on each of the ports for some period of

time in a round-robin fashion. In the current implementation, the monitoring takes place in a distinct phase as this is easier to implement and test for conformance. A commercial implementation would constantly monitor traffic and do the reconfiguration periodically, e.g. every day at mid-night, or when some threshold, e.g. the amount of routed traffic, is exceeded.

After receiving all the traffic matrices, the master switch controller calculates an overall traffic matrix and applies the variant of the greedy algorithm mentioned in Section 2 to partition the hosts into groups. For each group it then creates a VLAN description.

The VLAN description defines what characterizes packets as belonging to a given VLAN such that a switch can identify and forward them to the right destinations. What is needed is a mechanism for flow classification as proposed in the upcoming frame tagging standard IEEE 802.1q [8].

Because an implementation of this standard was not available by the time of our experiment we had to make use of a different mechanism which existed already on the switch. In Section 1 we listed the types of rules based on which the IBM 8277 Ethernet RouteSwitch is able to classify packets to VLANs. Remembering these rules, one realizes two possible rule types that allow to exactly identify the source of a packet: the MAC and the IP source address rule type. For convenience, we chose the latter because it allowed us to describe the set of machines that are members of a given VLAN by means of a single value: the IP subnet prefix.

Using a host's IP address to determine the VLAN with which the host should be associated, would seem to raise a problem: any set of hosts may form a communication cluster for a short period before breaking up and forming new clusters; the intention is to map such a cluster on the switches implementation of an IP rule-based VLAN, however it is not possible to allocate their permanent public IP addresses to enable them to belong to the appropriate VLAN as the appropriate VLAN varies over time and are created 'on the fly'.

The concept of a *private IP address* is used to solve this problem. When the master switch controller defines the VLAN, associated with that VLAN is a new IP subnetwork. After each switch controller receives the VLAN descriptions it informs its attached hosts about private IP addresses that they should use for communicating within that VLAN. A part of the IP addressing space is reserved for the creation of these dynamic IP subnets and the addresses are reclaimed each time the VLANs are updated. A host controller running on the Linux PC is informed about the effective mapping between the private and public address spaces. Based on this information a loadable Linux kernel module modifies the behavior of the original TCP/IP protocol stack such that it performs appropriate substitutions in inbound and outbound packets in a way that is transparent to applications. If a host recognizes that an IP address with which it communicates is now in the same VLAN as one of its own private addresses, it uses the private IP address in all future communications. All hosts remain members of the default VLAN and can still be reached via their public IP address across this VLAN. (This serves as a useful 'control channel' during debugging and for maintaining normal network infrastructure services such as DNS, NFS, etc.) The functionality of the add-on kernel module is almost equivalent to configuring new IP network interfaces atop the Ethernet device driver and adding routing entries for private IP addresses during the lifetime of dynamically created VLANs. (The actual code also must insert an arbitrary private MAC address in order to make the IBM 8277 RouteSwitch inspect the arriving frame; otherwise it would just rely on its cached forwarding decision.)

The overall effect is to reduce the amount of traffic that is carried across the large default VLAN by carrying some part of it over dynamically created small private VLANs. By introducing slight modifications to end points (hosts' protocol stacks) we were able to use a commercial switch whose capabilities cannot be extended by adding software. The concrete choice of using private IP address for marking VLANs is an implementation choice that is not substantial to our scheme, but was mainly motivated by pragmatic concerns. Other options could be based on alternative rules for describing VLANs to Ethernet switches (arbitrary bit patterns within frames, in the most general case). Having said this, we note that private IP addresses allow us to easily express cases where hosts belong to several VLANs at once, as will normally be the case with multiple applications causing different traffic flows (and requiring different levels of service discrimination). The design furthermore proved useful, because it kept implementation issues (such as the very existence of private IP addresses) transparent from client applications and human operators, while remaining compatible to a set of useful IP diagnostic tools for debugging.

In summary, a distinct VLAN control layer is introduced into the network whose entities collaborate in order to gather information about the current traffic patterns and determine which VLANs should be created. Information about the format and membership of the VLAN is passed between these entities — in our implementation in the form of private IP addresses.

### 4.1 Security

As well as increasing efficiency, VLANs also have a role in securing the network. They allow a well defined policy as to whom potentially sensitive traffic is broadcast. Automatically created VLANs need to fit into such a security policy. This can be achieved by enforcing security constraints which the switch controllers will take into account when creating the partition.

The current implementation addresses a similar problem in that our test network is also used as part of the laboratory's normal network infrastructure. It is necessary that normal communication is not impeded by experiments carried out over this network. In order to achieve this we have statically partitioned the network, and the switch controllers only act on one of these partitions.

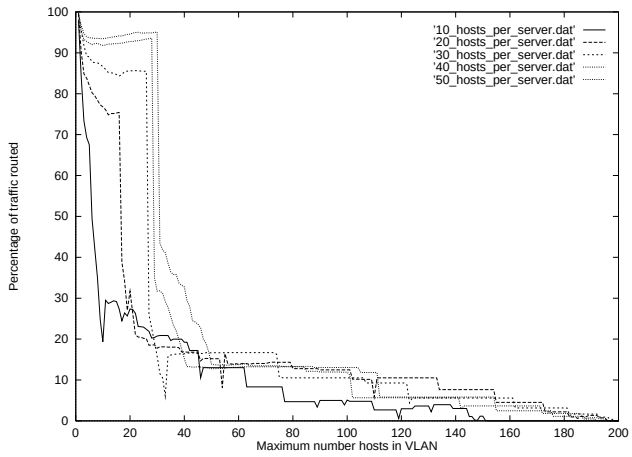
## 5 Simulation Results

This section presents the efficiency (reduction of routed traffic) and the performance (execution time) of the partitioning algorithm, measured by running it with different simulated traffic patterns. Simulating the network traffic allowed us to test the process in a variety of different configurations. The simulated networks all contained two hundred hosts corresponding to a single internet type C subnet.

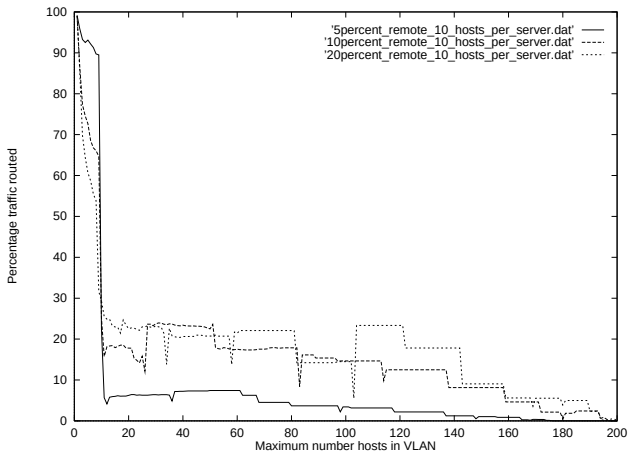
### 5.1 Client/Server Communication

In this simulation the traffic generators are a set of application servers and clients. Each client is associated with one and only one server. Ninety percent of traffic is between client and server, but the exact amount of traffic a given client communicates with its server is randomly chosen. The other ten percent of traffic is randomly allocated between arbitrary hosts (either clients or server).

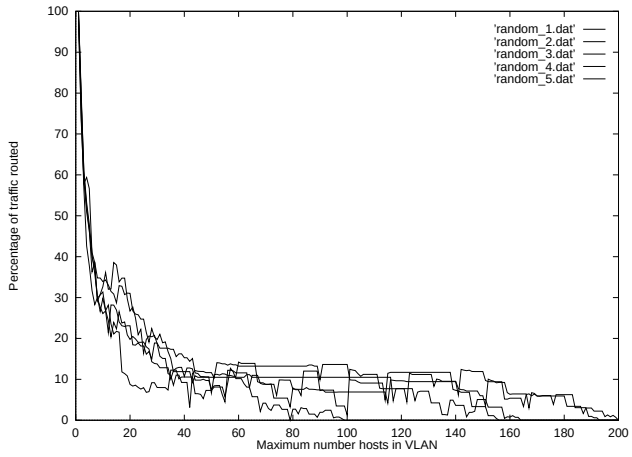
Figure 3(a) shows how the amount of routed traffic is reduced depending on the maximum number of hosts we allow to be in a



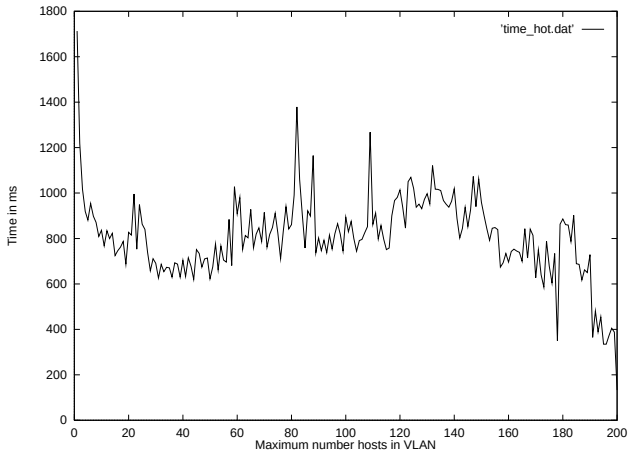

(a) Percentage of routed traffic for different cluster sizes.




(b) Percentage of routed traffic for different noise ratios.




(c) Percentage of routed traffic for random traffic patterns.




(d) Execution time of algorithm.

Figure 3: Quantitative results for automatic VLAN creation

VLAN for different numbers of clients per server. The percentage of traffic that is routed drops dramatically as the number of hosts per VLAN approaches the number of clients per server.

After that, further increasing the maximum number of hosts per VLAN only gradually reduces the amount of routed traffic (eventually reaching zero when all hosts are in the same VLAN). Increasing the VLAN comes at a cost in terms of amount of unnecessary broadcast traffic received; depending on the relative importance given to reducing the broadcast traffic and decreasing the routed traffic, the automatic VLAN creation system can come to decision about the best partition.

Figure 3(b) shows that the first part of the curves becomes flatter as the relative amount of non client/server traffic increases.

### 5.2 Arbitrary Peer-to-Peer Communication

The previous scenario motivates the correctness of the algorithm as it gives an easily predictable result: that servers and their client should be allocated to the same VLAN.

However, the intended context of our work is in environments where traffic patterns cannot easily be determined a priori and in consequence where static rule based VLANs break down.

Figure 3(c) shows the results for a scenario were every host communicates a randomly chosen amount of traffic with two other randomly chosen hosts. The clusters thus formed are impossible to predict. Five independent runs of the simulation are shown.

We can observe that the amount of routed traffic for this scenario can be reduced greatly already for a small maximum number of hosts per VLAN (between 20 and 40). The amount of routed traffic becomes zero when the maximum number of hosts per VLAN reaches the size of the biggest cluster. The results for the random

traffic patterns gives us some confidence that the algorithm would also behave correctly in a more dynamic environment.

### 5.3 Performance

The last figure (Figure 3(d)) shows the execution time of the algorithm for different maximum numbers of hosts per VLAN, using the client/server scenario. The execution times were measured on an IBM RS/6000 43P Model 140 workstation (332 MHz PowerPC 604e, 12.9 SPECint95) running AIX 4.3. The algorithm is implemented in Java and was run with the just-in-time compiler of JDK 1.1.7. The amount of time to distribute the data between the switches can be measured in some small number of milliseconds and is inconsequently compared to the time to calculate the partition.

Further work needs to be done looking at how long the network needs to be observed before reasonable predictions can be made. So for example, the network might be observed during an entire working day and reconfigured at night, or it might be that shorter, or longer periods are more suitable. However, Figure 3(d) shows that even our unoptimized system can calculate the partition for two hundred hosts in approximately a second in most cases and so the execution time of the calculating the partition is not a limiting factor.

## 6 Related Work

Other work on ATM 'cut throughs' has already been described in Section 3. To the best of the authors knowledge no existing VLAN scheme uses direct measurement as the means to create and assign hosts to Ethernet VLANs.

### 6.1 Products and Standards

[8] is a draft standard defining the architecture, services, and protocols that Ethernet switches should provide for VLAN support. It extends the Ethernet frame format to introduce a VLAN tag and priority; it specifies how the switch should allow interoperation between these extended Ethernet formats and the more conventional ones. These tags can either be statically defined by management (e.g. via SNMP) or through some more dynamic process. The VLAN tag plays the same role in [8] as the private IP address in our implementation. If and when such switches and network interface cards become available, then they could be used as the basis for distributing the VLAN descriptions generated from on-line measurement.

[15] gives a good overview of the general issue in VLAN technology; it mentions that (when written in 1996) VLANs, although popular with vendors, were less so with users due to a perceived complexity in their management.

Ethernet products such as [7, 1] use rule based systems to determine membership of already existing VLANs. Information about the capabilities of products is readily available through the vendors web sites.

### 6.2 Ethernet QoS

Observed on a long enough time scale, Ethernet divides the channel equally between attached hosts. Due to the increasing need for continuous media service to the desktop there is an interest in allowing a host to obtain some guaranteed amount of the channel (or Quality of Service, for example [16, 17]). The strategies proposed all involve enhancing the normal Ethernet control mechanism in order to determine who should send when, at the cost of making it more complex. For example [17], adds a demand priority protocol to allow hosts to signal the importance of their traffic to an Ethernet hub. A detailed examination of this work is beyond the scope of this paper; however it should be noted that the addition of additional control mechanism to Ethernet (e.g. the dynamic distribution of VLAN/QoS tags) is complementary to the work described here.

### 6.3 Measure

The Measure project [9] has proposed ways in which the effective bandwidth of a switch can be determined by measuring the traffic flowing through it; this avoids traffic emitters having to try and characterize their arrival process at the switch in terms of the ATMF traffic parameters [18] (e.g. peak rate, burst size) which is difficult to do a priori for a wide class of continuous media services. Measure call acceptance control accepts a bursty flow at its peak rate, observes its for a period, and determines the actual effective bandwidth it will require in the future, taking advantage of the self-similar nature of the traffic. The work in this paper is analogous to the Measure work in that it avoids having to characterize network behavior a priori but observes what happens in practise. At the moment the automatic creation of the VLANs does not make use of any of the complex mathematics used within the Measure, but there may be scope to introduce it when the resource reservation issues are addressed.

## 7 Conclusion

This paper has proposed on-line measurement as the means to determine the appropriate VLANs that should exist on a network. This allows a system which is responsive to change and is appropriate for LANs in which the traffic patterns are unpredictable and variant. This paper has motivated the practicality of the technique through the description of a working system in which this technique is used.

## References


[1] IBM, "IBM 8277 Ethernet Switch," *IBM release notes*, 1999.

[2] J. Martillo, "Routing in a Bridged Network," *Telford Tool inc. white paper*, 1990.

[3] R. Ciampa, "Flexible Intelligent Routing Engine (FIRE)," *3COM White Paper*, 1997.

[4] P. Newman, G. Minshall, and T. Lyon, "IP Switching: ATM Under IP," *Submitted to IEEE/ACM Transactions on Networking*, 1997.

[5] P. Newman, G. Minshall, T. Lyon, and L. Huston, "IP Switching and Gigabit Routers," *IEEE Communications*, vol. 35, pp. 64–69, January 1997.

[6] Y. Rekhter, B. Davie, D. Katz, E. Rosen, and G. Swallow, *Cisco System's Tag Switching Architecture Overview*, February 1997. RFC 2105.

[7] Cisco, "Catalyst familiy of LAN Switching," *Cisco System Inc. product information*, 1998.

[8] IEEE/ISO/IEC, "Virtual Bridged Local Area Networks," *ISO Publication*, July 1998. Draft Standard: IEEE Standard for Local and Metropolitian Area Networks, P802.1Q/D11.

[9] S. Crosby, I. Leslie, M. Huggard, J. Lewis, B. McGurk, and R. Russell, "Predicting Bandwidth Requirements of ATM and Ethernet Traffic," in *Proceedings of IEEE 13th UK Teletraffic Symposium*, (Strathclyde University, Glasgow), March 1996.

[10] P. Crescenzi and V. Kann, "A compendium of NP optimization problems," *Available as appendix to book: Complexity and Approximation, Combinatorial Optimization Problems and their Approximability Properties by Ausiello et al* , August 1998. Springer-Verlag.

[11] ATMF, "LAN Emulation over ATM Specification, Version 1.0," *The ATM-Forum: Approved Technical Specification*, 1995.

[12] ATMF, "Multi-Protocol Over ATM (Version 1.0)," *The ATM Forum Technical Committee, af-mpoa-0087.000*, July 1997.

[13] Luciani, Katz, Piscitello, and Cole, "NBMA Next Hop Resolution Protocol (NHRP)," *Draft internet standard*, October 1997.

[14] S. Waldbusser, "Remote Network Management Information Base," *Draft Standard, Internet RFC 1757*, October 1995.

[15] D. Passmore and J. Freeman, "The Virtual LAN Technology Report," *3COM White Paper*, 1996.

[16] J. L. Sobrinho and A. Krishnakumar, "EQuB — Ethernet Quality of Service Using Black Burst," *Local Computer Network Conference*, 1998.

[17] P. Kim, "Deterministic Service Gurantees in IEEE 802.12 Networks - Part I: The Single-Hub Case," *IEEE/ACM Transactions on Networking*, vol. 6, pp. 645–658, October 1998.

[18] S. S. Sathaye, "ATM Forum Traffic Management Specification Version 4.0," in *ATM Forum Technical Committee - Contribution 95-0013*, 1995.