An Architecture for Automatic Configuration of Integrated Networks

Salvatore D'Antonio[†], *Maurizio D'Arienzo*[†], *Antonio Pescapè*[†], *Giorgio Ventre*[†]
[†] *Dipartimento di Informatica e Sistemistica - Università di Napoli "Federico II" (Italy) – {maudarie, pescape, giorgio}@unina.it*
[†] *ITeM - Laboratorio Nazionale CINI per l'Informatica e la Telematica Multimediali (Naples, Italy) - {salvatore.dantonio}@napoli.consorzio-cini.it*

Abstract

Configuration and management activities are frequently performed both in Local Area and Campus Networks due to the intrinsic variability characterizing such networks as well as innovative services provided by means of them. Indeed, in order to benefit from services like Voice over IP and multimedia content distribution, corporate users need to appropriately configure and manage their network. Suitable strategies have to be undertaken to fulfill stringent requirements imposed by such services on the underlying "integrated" transport infrastructure. These activities are both time and money consuming since they usually are under the responsibility of network administrators and managers. In this paper we present an architecture that allows the configuration of network devices in an automatic fashion in order to facilitate traffic management and prioritization in LANs. On one hand, traffic management is optimized through the segmentation of a corporate network in multiple Virtual LANs via SNMP protocol. On the other, traffic prioritization is carried out by grouping LAN packets into separate classes associated with different priority levels in compliance with 802.1p. The segmentation process is carried out in two steps: in the former the network segmentation in "multimedia hosts" (i.e. IP Phone and Multimedia PC) and "data hosts" is accomplished (as well as traffic prioritization), in the latter the segmentation task is optimized in both VLANs thanks to the utilization of a "Partitioning Algorithm".

Keywords

Integrated Network, Automatic Network Management and Configuration, SNMP, VLANs.

1. Introduction

Integration currently represents the most innovative concept and, at the same time, the most difficult challenge for all network designers: a single network infrastructure for the transport of all the traffic (data, voice and video). Not only a single network infrastructure but also a single protocol, i.e. the IP protocol, glue of all applications on different platforms: situations in which both wireline world and

0-7803-8230-7/04/$20.00

wireless world are melted together are nowadays realities and the current trend is the definition of a global communication paradigm, independent of the particular scenario. For corporate users (universities, small and medium enterprises) exploiting the same network to transport voice, video and data represents a good solution to common issues, such as management of different platforms and knowledge of distinct technologies. To make it feasible, a proper configuration activity has to be performed. Indeed, in order that a corporate network may support both data and real-time communications, traffic separation and prioritization have to be ensured by appropriately configuring involved network devices. Therefore, configuration has become a critical requirement for managers in today's highly interoperable networks. Up to now, vendor specific mechanisms to transfer configuration data to and from a device have been developed. However, each of these mechanisms may be different in various aspects, such as session establishment, configuration data exchange, and error responses. On the other hand, several utilities are used to control devices via CLI (*Command Line Interface*), but they are prone to failure due to the instability and lack of uniformity inherent in a CLI. Such considerations highlight the need for tools based on standard solutions and enabling device configuration in an automatic way.

In this paper we present an innovative approach to traffic management in Local Area Networks: traffic separation and prioritization are ensured thanks to a network segmentation activity in which the SNMP standard protocol [1] is used to configure each device. Basically, by exploiting the functionalities of such a protocol two distinct Virtual LANs (*VLANs*) [2] [3] [4] [5] are created in order to separate real-time traffic from data traffic over a single network. For this purpose, a preliminary network monitoring activity is performed so as to identify different traffic sources and to help the VLAN configuration process. Indeed, our aim is to classify network traffic into two separated classes: real-time traffic, associated with multimedia applications (videoconferencing, Voice over IP, etc), and data traffic, referring to any other traffic. In order to distinguish real-time traffic from other traffic, the traffic analysis activity is appropriately designed so that RTP (*Real Time Protocol*) [6] packets are filtered. At the end of this phase, two broadcast domains are dynamically configured via SNMP on the basis of information gathered during the traffic analysis activity: a *Data VLAN* and a *Voice VLAN*. The Voice VLAN comprises multimedia PCs, IP phones and any other network host sending RTP packets. The network segmentation, carried out through the configuration of two different VLANs, represents the first step towards the traffic prioritization. In fact, packets arriving at switch ports corresponding to real-time traffic sources are tagged with a higher priority level, in accordance with 802.1p. After this first segmentation a network optimization process is performed: for each VLAN ("Data" and "Voice") a monitoring process determines real communication both among hosts and multimedia PCs and IP phones. In this way, we can group together only the "stations" that really communicate. A real example of Local Area Network, whose configuration is managed by following the proposed approach, is depicted in Figure 1. Experimental results reported in this paper are obtained by using as network testbed the "Computer Science Department and ITeM Lab"

integrated network. Such network is one of the real scenarios where our architecture is currently running [7].

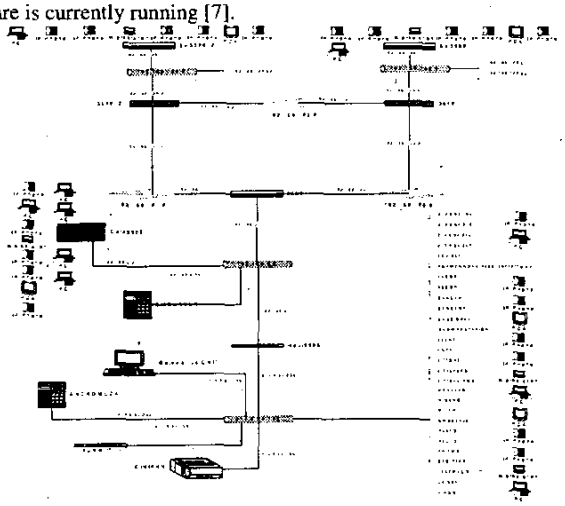

Figure 1: The Computer Science Department and ITeM Lab network

The paper is organized in 5 sections. After this introduction, Section 2 illustrates the context where our contribution has to be positioned and deals with related work. In section 3 "Dynamic VLAN configuration and priority setting" software architecture aiming at configuring and managing LANs in an automatic fashion is presented. Section 4 illustrates the "VLAN manager" software architecture as well as the "Partitioning Algorithm". Finally, section 5 provides some concluding remarks, together with some directions of future work.

2. Motivation and related work

The network management is the collection of tasks performed to maximize availability, performance, security and control of a network and its resources [8] [9]. The International Organization for Standardization (ISO) Network Management Forum has divided network management into five functional areas: Fault Management, Configuration Management, Performance Management, Accounting Management, Security Management. Configuration management deals with initialization, modification and shutdown of a network device [10] [11]. Networks are continually adjusted when devices are added, removed, reconfigured, or updated. These changes may be intentional, such as adding a new switch or hub to the network, or path related, such as a new connection between two devices resulting in a re-routed path. If a network is to be turned off, then a graceful shutdown in a prescribed sequence is performed as part of the configuration management process. The process of configuration management involves

identifying the network components and their connections, collecting each device's configuration information, and defining the relationship between network components [12]. .Configuration tasks, such as installing or reconfiguring a system, provisioning network services and allocating resources, typically imply a large number of activities involving multiple network elements. Furthermore, heterogeneity of network devices increases the complexity of configuration and management activities. Without automatic configuration and management, IT managers and administrators must manually configure and manipulate network devices through CLIs: this process is both error prone and time consuming. This approach also leads to difficulties in making adjustments to network parameters, even if network usage patterns are reasonably knowable over time. Given these complexities, dynamic network configuration based on usage requirements can only be achieved with some type of automatic management system. And that's where our approach and our architecture can help. Figure 2 shows cumulative results from several large enterprises and Internet Service Providers (ISP) that describe the commonly occurring problems encountered when a device configuration is changed. It identifies five general categories of network configuration problems, which are defined as follows:

- Process is a general group that includes errors in obtaining the right approval, applying the right configuration to the wrong device, applying the right configuration at the wrong time, and other similar problems.
- Partial configuration means that only a part of the intended and complex change was correctly applied.
- Mis-Configuration means that there was an error in the applied configuration.
- Security means that a change was needed to neutralize a hacking, virus, or other security related problem.
- Induced Change category means that a subsequent change was required from a previous configuration change.

It was estimated that about half of all changes to the network result in some kind of reported problem elsewhere on the network. Considering that large networks could have more changes daily, an automatic process that will eliminate these errors is greatly needed [13]. Therefore, based on such considerations the availability of tools enabling to configure network devices in an automatic fashion represents an important benefit for network managers.

In the following we present some existing solutions aiming at addressing issues related to configuration and management of network devices. In [14] the design of an SNMP management tool called "scli" is described. Such a tool provides an easy and effective way to use SNMP command line interface in order to configure Virtual LANs, but it differs from our approach since the VLAN configuration is not based on a preliminary network monitoring activity and, therefore, the network segmentation process is not completely dynamic. In [15], the authors present Java ADministration system (JAD). It is a Java based software for the configuration and management of network devices using a web browser. It is a simple but powerful tool that allows users to manage a network device via any Java enabled Web browser. In [16] configuration of network elements is managed

via a set of high-level rules or business policies rather than device-by-device. A network management model for multimedia services in a broadband wireless access network (BWA) is illustrated in [17]. In such a model, SNMP is exploited for both configuration management and fault treatment. In [18] an Automatic VLAN creation based on on-line measurement is presented: our work follows the same approach with the additional feature of presenting an implementation on real networks. Finally in [19] a VLAN Management System is presented.

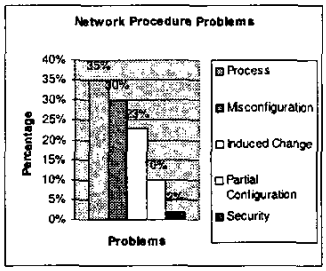

Figure 2: Network procedure problems [13]

3. Dynamic VLAN configuration and priority setting

In this section we present the details of our architecture for automatic VLAN and priority configuration in order to support real time traffic over Multiservice Local Area Networks. This architecture is capable to increase the performance of an "Integrated Network" [7] [20]. This work starts from others similar approaches [21] [18], but it is focused on real time traffic separation and prioritization in real networks. Our software architecture is built of the following main modules: i) Network Monitoring Module; ii) Data Analysis Module; iii) VLAN Configuration Module; iv) Priority Configuration Module. These functional blocks are related to four phases: such phases are shown in Figure 3. Before describing the software modules composing our architecture, we aim at underlining a relevant aspect. We have implemented such modules based on CISCO network devices. It has to be noticed that configuring VLANs comprising CISCO IP phones is already possible in networks built of CISCO devices. In this work we present a global approach suitable to manage network architectures including both Multimedia PC and CISCO IP phones. Moreover, the proposed architecture is extensible to any network device: in this section the open and heterogeneous model our architecture is based on is clearly illustrated. In order to ensure the right behavior of the overall architecture it is necessary, as first step, to connect the Management Station to the switch port associated with *Management VLAN* and to configure it as *port monitoring*. In a real implementation of our architecture, the reference network is built of CISCO Catalyst 3500 XL (C3524 PWR XL EN) switches. In order to configure them, the following MIBs (Management Information Bases) are necessary: the *BRIDGE-MIB*, the *CISCO-VLAN-MEMBERSHIP-MIB* and, finally,

the *CISCO-VTP-MIB* [22] [23]. The first one is introduced in RFC 1493 - this document defines the section of MIB related to the management of 802.1D-compliant MAC-bridges - ({internet(1) mgmt(2) 1}), whereas the others are vendor-dependent ({internet(1) private(4) 1}, CISCO sub-tree). In case a diverse switch has to be managed, the only needed change is to use different MIBs.

3.1. Network Monitoring Module

The goal of the Network Monitoring Module is to identify what network devices are communicating and what traffic they are generating. In fact, the Network Monitoring Module produces a list of the IP addresses corresponding to the network hosts, each labeled as data or real-time traffic source. With reference to software details, this module is based on WinPcap library [24] [25].

3.2. Data Analysis Module

Taking into account the output of the Network Monitoring Module, the Data Analysis Module performs the following task: for each network host, it infers the association between the MAC address and the number of the switch port to which the host is connected.

3.3. VLAN Configuration Module

Information provided by the Data Analysis Module represents the input to the VLAN Configuration Module. Such à module is responsible for the creation of two VLANs: the Voice VLAN and the Data VLAN. On the basis of the previously determined associations between switch ports and traffic sources, the VLAN configuration Module segments the network in two parts by using SNMP commands, such as GET, WALK and SET. Data Analysis Module provides two lists of hosts: the first list contains data traffic sources whereas in the second list there are multimedia traffic sources. In these lists, for each host, both IP address and MAC address are specified. During the configuration process via SNMP it is necessary to know the association between MAC addresses and switch ports by checking the OIDs (Object Identifiers) of the used BRIDGE-MIB. To this purpose, the module sends ICMP Echo Request packets to listed hosts: this operation is intended to start the process, called backward learning. Since the final part of the OID is the decimal representation of the MAC address, the associations between MAC addresses and switch ports can be easily determined by executing an SNMP-WALK command. In fact, this command returns, for each OID, an integer value n from which the switch port number can be inferred taking into account the following rule (such associations are present in a typical CISCO scenario and they are vendor dependent): if the switch port $\in$ (1, 8), then $n \in$ (13, 20); if the switch port $\in$ (9, 16), then $n \in$ (22, 29); if the switch port $\in$ (17, 24), then $n \in$ (31, 38). In order to determine such associations our architecture imposes the following requirement: the network must be "full switched" (In our context this statement means that there is a "1 to 1" association between the switch port and the host). Only in this case it is possible to separate the network traffic. However, if there are both data traffic sources and real time traffic sources connected to the switch port

"x" by means of an hub, such a port can be assigned to either the data traffic sources list or the multimedia traffic source list (by default, the port "x" is assigned to this last list). After giving a general view of the proposed architecture, now we focus on implementation details.

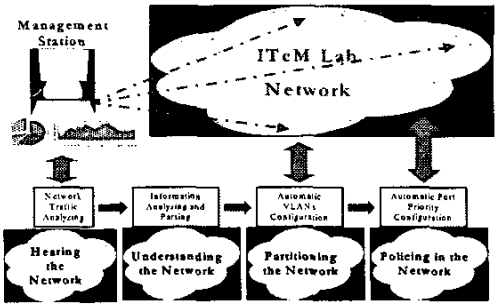

Figure 3: Dynamic VLAN configuration and priority setting: reference model and architecture

The first step consists in assigning every switch port to the Management VLAN. After this operation, called *set_allPORT_VLANdefault,* the Bridge MIB contains information about all switch ports. Thanks to the output of traffic sniffing phase, the router (switch layer 3) address can be identified. Hence, the corresponding switch port is configured as *Multi-VLAN* port (or as *Trunk* port). Since Multi-VLAN ports belong to each configured VLAN, they enable the communication among different VLANs and the connection to the Internet. However, such ports cannot be associated to the Data VLAN or the Voice VLAN and their status cannot be modified. At this point, by using the CISCO-VLAN-MEMBERSHIP-MIB an SNMP-GET command is executed in order to retrieve information about status of each switch port. In particular, such information is determined by analyzing the value of each OID. In this case, Object Identifiers have the following format: 1.3.6.1.4.1.9.9.68.1.2.2.1.1.i where "i" is an integer value - $i \in (2, 25)$ - identifying the "i-1"–th switch port. After gathering information about status of involved ports, the next step consists in determining VLANs already configured. In order to obtain such information, an SNMP-GET command, having as parameter the CISCO-VTP-MIB, is executed. After this operation, names and identifiers of existing VLANs are available: two of them are configured as Voice VLAN and Data VLAN. The last step is the VLAN configuration process that is automatically performed. On the basis of information previously determined, such as identifiers of already existing VLANs and associations among traffic protocols, MAC addresses and switch ports, the following Virtual LANs are configured: Voice VLAN (comprising switch ports corresponding to RTP packets sources); Data VLAN (comprising data traffic sources); Management VLAN (including the management station); Multi VLAN. By default, the first two existing VLANs are, respectively, labeled as Voice VLAN

and Data VLAN. In order to assign each switch port to the right VLAN, it is necessary to perform an SNMP-SET operation requiring the following parameters: the VLAN identifier and the OID that is 1.3.6.1.4.1.9.9.68.1.2.2.1.2."j"+1 where "j" identifies the switch port. At the end of this task a complete report containing information about each configured VLAN is created.

3.4. Priority Configuration Module

The last module, called Priority Configuration Module, is in charge of providing the real time traffic with specific guarantee in terms of network performance. Such a phase aims at exploiting a useful feature characterizing most switches currently commercialized, i. e. the compliance with the 802.1p. Thanks to the 802.1p standard, LAN switching has taken a step forward in the move towards supporting the convergence of voice, video and data. 802.1p is a specification for giving Layer 2 switches the ability to recognize the packet priorities. To be compliant with 802.1p, Layer 2 switches must be capable to group incoming LAN packets into separate traffic classes. The specific value associated with each switch port indicates the priority level applied to untagged packets arriving at that port. Eight classes are defined by 802.1p. On a congested network that uses 802.1p extension, a packet with higher priority receives preferential treatment and is served before a packet with a lower priority. The header field, named Class of Service, of an Ethernet frame can be set with the value representing the priority level which such a frame belongs to. The highest priority is seven and it is usually reserved for network-critical traffic. The zero value is used as a best-effort default, invoked automatically when no other value has been set. Priority packet tagging enables to differentiate traffic treatment based on priority level. In our case, in order to give precedence to real time traffic over best effort traffic, switch ports can be set with different priority value depending on type of traffic arriving at them. In particular, we introduce two different priority levels, respectively for "VLAN Voice" and "VLAN Data". In Figure 4 two different scenarios intended to show benefits caused by traffic separation and network segmentation are depicted.

4. A module for VLAN management: VLAN Manager

After the initial configuration of both VLANs (Voice and Data), the second step is represented by a further segmentation performed by exploiting a Partitioning Algorithm. Such segmentation allows gathering all active hosts and IP phones in the same VLAN. By assuming that hosts keep on communicating in the future [26], the aim of the application is to create a VLAN for each group of communicating hosts. Current tools are based on static information to assign the pertaining VLAN. Conversely, our application relies on *on-line* measures *and* computations. This feature enables efficient assignment of hosts to VLANs according to the current traffic situation.

4.1. Automatic sniffing module

Sniffing activities help to build "Traffic Table" and "Hosts/IP Phones Table". The

first one (traffic table) contains information about "*who is talking to whom*", the second one (device table) the total amount of traffic generated by every single station. These actions are automatically performed.

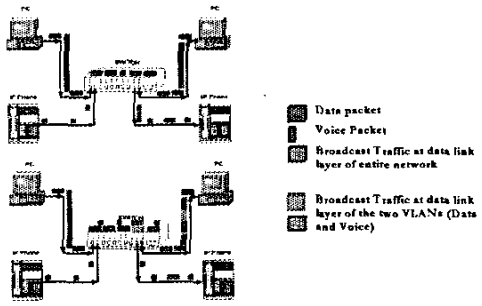

Figure 4: Data traffic and Voice traffic in two separated domains

4.2. SNMP Discovery module

"Traffic Table", which is the output of previous module, contains all measures concerning traffic flowing towards all network devices, included switches and routers, as well as both broadcast and multicast traffic. Due to the need to analyze communications among the hosts, a *Discovery* module is introduced in order to discover all the computers active in the network. Such a module is able to distinguish devices, and it returns addresses of switches and routers to be deleted from the final table. During this process, a supplementary table is created: the "Address Table". This table gives the association between IP addresses and MAC addresses. As an example, in case of 254 hosts (PCs and IP Phones), a square matrix with null diagonal is built. In a 192.168.128.0 network, the element $a_{32,111}$ represents the traffic between 192.168.128.32 and 192.168.128.111.

4.3. Partitioning Algorithm module

The *Partitioning Algorithm module* configures VLANs starting from traffic measurements. In case of application on other matrixes, as added value, the Partitioning Algorithm computes the best "Maximum Group" (*Maximum Clique*) [27][28][29] among all matrix elements. The algorithm creates VLANs on the basis of traffic flowing among network stations belonging to a Campus or Local network. This fact requires the knowledge of the status of all networked stations, in particular whether each pair of stations belongs to the same group: "Threshold Traffic" (and its calculus) is the parameter that gives the information concerning the correlation between two stations. Under normal circumstances this parameter is automatically calculated (e.g. as mean value), but it is also possible to set it in a manual way or even to let it come out from a closed-loop algorithm. In this way, a more flexible approach can be taken, since the network manager has full control on the threshold whose value has a great impact on the overall algorithm results. We

added this possibility for our experimentation, too. Stemming from the analysis of measures carried out in laboratory, we choose the mean value of traffic matrix elements as automatic parameter. It is important to notice that, during the threshold computation, a variation in the threshold causes a variation in number of VLANs, but there is not a proportional correlation between them. A bigger threshold does not roughly imply a lower number of VLANs. This behavior can be explained by observing the variation in communication matrix with respect to the variation in threshold, which causes a modification in Maximum Groups. Different criteria have been adopted for the threshold. While in a first time a multiplier factor has been chosen, in the following a new simpler and more correct method has been preferred: after the automatic calculus of mean value, the sorted vector containing traffic values is inspected until the number of VLANs fits the desired value. First step in the algorithm is the achievement of Traffic Matrix. For instance, for C class addresses, the matrix is 254x254, in case of B class addresses the matrix will be 65534x65534. Traffic Matrixes are usually sparse, that suggested us the use of "*Sparse Matrix*" and of "*Three Vectors Method with pointers*" methods. After the Matrix acquisition, the threshold is computed. Then, elements that overtake threshold are considered and the matrix is modified in a Triangular Boolean Matrix that has "1" where elements overtake threshold, while "0" in other parts. Of course, because the Matrix is triangular, elements a_{ij} and a_{ji} are equal. A matrix built in this way represents an *Undirected Graph*.

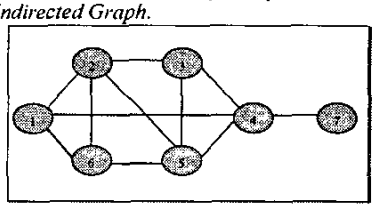

Figure 5: Test Graph for Partitioning Algorithm

The example depicted in figure 5 describes the subsequent phase of construction of "Maximum Group". In this case, as far as figure 5, Maximum Groups are: VLAN 1 : 1, 2, 6; VLAN 2 : 2, 3, 5; VLAN 3 : 2, 5, 6; VLAN 4 : 3, 4, 5; VLAN 5 : 4, 7; VLAN 6 : 4, 1. Group 2,3,5,6 is not present because there are no communications between 3 and 6, thus two separated groups are created. Group {3,4} is not present because it is included in a bigger group. Group {4,7} is a stand-alone group. The Algorithm works in this way: from the first node (selected in a random choice), the whole tree is then checked. A binary tree is created so that only two decisions are possible: Left: current group without last node appended; Right: current group with last node appended. Every new append requires a check on the group, that means a check on every station to understand if it talks to all other stations of the group. When no other decision can be taken, current group is stored, and that group represents a "Maximum Group". The process starts again when the process on the last node is completed. Figure 6 should clarify the last part of the described process (The process is described with respect to only some nodes.

The complete tree has bigger dimensions, in spite of a test graph comprising just 7 nodes. Hatch indicates other routes to be followed.). In figure 7 the description in *Pascal like* notation is reported.

4.4. Automatic Setting Module

Finally, *Partitioning Algorithm* output enters the *Automatic Setting* module. Automatic Setting module follows the same approach presented in sections 3.1-3.3, where we present VLAN creation and configuration in case of Voice and Data. In this context the Automatic Settings Module configures switches by using SNMP.

5. Conclusions and future work

Most network designers have to balance the trade-off between existing implementations and the up-grades due to technological innovations, having, at the same time, to cope with the problem of guaranteeing continuity in the service provided by the networks. Among the main issues, the coexistence of different protocols plays a major role; if underestimated, it can bring the whole architecture to work in sub-optimal conditions and, in the worst case, even to starvation. VLANs are the natural answer to such problems. Apart from the large offer of proprietary solutions, provided by a number of vendors, a standard protocol, named 802.1q, currently exists. The VLAN paradigm represents a mean for the provision of many important features, such as security, virtual working groups, controlled resource sharing and broadcast domains reduction, in front of an increase in the load associated to configuration and management tasks.

In this paper we presented a novel approach to the design and development of a flexible architecture for the effective (and automatic) configuration and management of Local Area Networks. Starting from the concept of "integration", we pointed out the need for a global architecture enabling traffic separation and prioritization in an automatic fashion for LANs supporting both data traffic and real-time traffic. We then illustrated a model based on a modular decomposition of tasks involved in the network configuration process. This work presents a real implementation that actually proves the benefit of the automatic configuration management process by means of SNMP. On this topic an IETF group is started [32]. Segmenting a corporate network in two VLANs allows reducing the broadcast traffic generated by multimedia applications (VoD, VoIP, etc). Due to paper space limitation, we use this parameter to validate the architecture. In Figure 8 the percentages of broadcast and unicast traffic before and after automatic VLAN configuration in our real network (Figure 1) are depicted. In the first diagram the traffic from multimedia devices (Multimedia PCs and IP Phones) compared with the total measured broadcast traffic is reported. The second diagram refers to the traffic generated by the multimedia devices after the configuration of the Voice VLAN. It has to be noticed that such results are strongly dependent on both network equipments composing the reference testbed and used operating system. In addition, beside to intuitive consideration, by using experimentations on real networks it is possible to demonstrate the goodness of our approach. End-to-End

delay, jitter and throughput can be used like output parameters. In our real experimentations, in order to study the benefit of our proposal approach, we envisaged the use of the delay measurement of different network configurations and, in the case of voice traffic, the use of the MOS (*Mean Opinion Score*) parameter. First output results demonstrate the useful of our idea. We will show these results in a future work.

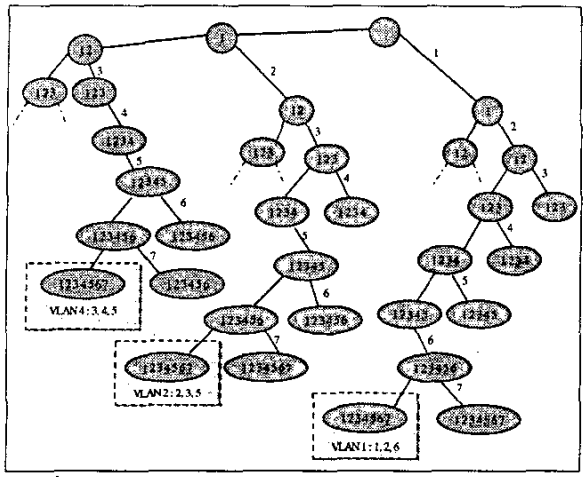

Figure 6: Solution Tree of Graph in figure 5

```
boolean get(TListVertix Curr-Whole, TVertix Vertix)
begin
  if Curr-Whole.Admissible then
  begin
    if (ExistNextVertix(Vertix)) then
   begin
   NextVertix:= NextVertix (Vertix)
      right = get(Curr-Whole + NextVertix, NextVertix);
      left = get(Curr-Whole, NextVertix);
      if (not right and not left) Mem(Curr-Whole);
      return true;
   end
    else begin
      Mem(Curr-Whole);
      return true;
    end;
  end else
  begin
    return false;
  end;
end;
```

Figure 7: Solution Procedure in a Pascal-like language

As far as the future work, testing the architecture in networks composed of devices from different manufacturers in order to evaluate its performance represents an important issue. Current implementation is strictly linked to features

of CISCO devices [30], composing the network testbed. However, the software architecture can be easily adapted to network scenarios in which other devices are utilized. Currently, we are testing our architecture by using network devices from manufacturers, like 3Com [31] and D-Link. In these cases only one change is needed: the network administrator must substitute the vendor switch MIB. Once this change has been made, the entire architecture keeps on working without problems. As for real time traffic, experimental results witness the effectiveness of our architecture: the creation of distinct VLANs ensures good network performance to such traffic.

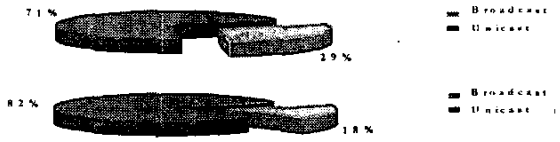

Figure 8: Broadcast Traffic reduction

Furthermore, it is clear that a fundamental issue is related to the behavior of the proposed architecture in large-scale scenarios. Indeed, scalability represents one of the most important aspects that should be emphasized when implementing multiservice networks. A sharp analysis of potential bottlenecks of our architecture is needed, with the respect to both the single modules and their orchestrated operation. We aim to evaluate the relationship between the VLAN configuration time and the network size. Thus, our future goal is to setup a certain number of networks of different size so as to measure the time needed to carry out traffic separation and prioritization in different network scenarios. Future network testbed will include heterogeneous mobile devices.

Acknowledgement

This work has been carried out partially under the financial support of the MIUR in the framework of the WEB-MINDS FIRB Project "Middleware for advanced services over large-scale, wired-wireless distributed systems".

References

[1] J.D. Case, M. Fedor, M.L. Schoffstall, C. Davin, "Simple Network Management Protocol (SNMP)", RFC 11157, May 1990

[2] RFC 3069 "VLAN Aggregation for Efficient IP Address Allocation" http://www.faqs.org/rfcs/rfc3069.html

[3] IEEE, "Draft Standard P802.1Q/D11, IEEE Standard for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks," July 30, 1998, ftp://p8021:-go_wildcats@p8021.hep.net/8021/q-drafts/d11/q-d11.pdf

[4] IEEE 802.1Q Virtual Bridged Local Area Networks, http://grouper.ieee.org/groups/802/1/vlan.html

[5] Virtual LAN Defined, http://www.networks.digital.com/dr/envisn/env5.html

[6] IETF Audio-Video Transport Working Group. "RTP: A Transport Protocol for Real-Time Applications", RFC 1889, January 1996
[7] A. Pescapè, S. D'Antonio, G. Ventre, "Un'applicazione per il Supporto del traffico real time su reti IP", AICA 2002, - pagg. 645-659 – September 2002, Conversano (Italy)
[8] W. Stallings, SNMP, SNMP v2, and CMIP: The Practical Guide to Network Management Standards, Addison-Wesley, Reading, MA, 1993.
[9] W. Stallings, SNMP, SNMPv2, SNMPv3, and RMON 1 and 2, Addison-Wesley, Reading, MA, 1999.
[10] M. Rose, The Simple Book: An Introduction to Internet Management, Revised Second Edition, Prentice Hall, Englewood Cliffs, NJ, 1996.
[11] M. Subramanian, Network Management: Principles and Practice, Addison-Wesley, Reading, MA, 2000
[12] Western Multiplex, "Implementing SNMP in a Microwave Radio Network", Technical Information, March 2001
[13] J. Strasser, "A New Paradigm for Network Management: Business Driven Device Management", July 2002
[14] J. Schonwalder, "Specific Simple Network Management Tools", LISA 2001,
[15] M. Chirico, F. Giudici, S. Sandolo, A Sappia, A.M. Scapolla, "The JAD System Architecture"
[16] S. Boros, "Policy-Based Network Management with SNMP", August 2000
[17] You-Sun Hwang, Eung-Bar Kim, "The Management of the Broadband Wireless Access System with SNMP", 10th ICT 2003, February 2003
[18] S. Rooney, C. Hörtnagl, J. Krause, "Automatic VLAN creation based on on-line measurement", ACM SIGCOMM 1999, pp: 50 - 57, Volume 29 - Issue 3 (July 1999), ISSN:0146-4833
[19] T. Miyamoto, T. Tamura , R. Suzuki , T. Hiraoka , H. Matsuo , M. Izumi , K. Fukunaga, "VLAN Management System on Large-scale Network", IPSJ JOURNAL "Special Issue on Construction and Management of Internet Application Systems" Vol.41 No.12.
[20] A. Pescapè, G. Ventre, G. B. Barone "Vlan Manager: An Architecture for Automatic VLAN design, configuration and management", AICA 2000, pagg. 327-342.
[21] ftp://ftp.cisco.com/pub/mibs/supportlists/wsc3500xl/wsc3500xl-supportlist.html
[22] http://www.cisco.com/univercd/cc/td/doc/product/lan/cat5000/rel_4_1/mib/mover.htm
[23]Analyzer, http://www.netgroupserv.polito.it/analyzer
[24] Ethereal, http://www.ethereal.archive.sunet.se/distribution
[25] UCD-SNMP Project 4.2.1, http://www.net-snmp.sourceforge.net
[26] S. Crosby, I. Leslie, H. Huggard, J. Lewis, B. McGurk e R. Russel, "Predicting Bandwidth Requirements of ATM and Ethernet Traffic", IEEE 13th UK Teletraffic Symposium, Mach 1996
[27] E. Balas, C. S. Yu, "Finding a Maximum Clique in an Arbitrary Graph", SIAM J. Computing, Vol. 15, No 4, 1986.
[28] I. M. Bomze, M. Budinich, P. M. Pardalos, and M. Pelillo, "The Maximum Clique Problem", Handbook of Combinatorial Optimization, vol. 4, Kluver Academic Publisher, Boston Pattern MA, 1999.
[29] C. Bron and J. Kerbosch, "Finding All the Cliques in an Undirected Graph", Communication of the Association for Computing Machinery 16 (1973), 575-577.
[30] Cisco VLAN Roadmap, http://www.cisco.com/warp/public/538/7.html
[31] D. Passmore, J. Freeman, The Virtual LAN Technolog Report, March 7, 1997, http://www.3com.com/nsc/200374.html
[32] M. MacFaden, D. Partain, J. Saperia, W. Tackabury, R. Rajan, A. Sastry, "Configuring Networks and Devices With SNMP", draft-ietf-snmpconf-bcp-09.txt, June 2002