Building the Network Infrastructure for the International Mathematics Olympiad

Rudi van Drunen, Competa IT, (R.van.Drunen@competa.com),
Karst Koymans, University of Amsterdam, (K.Koymans@uva.nl)
The Netherlands

Abstract

In this paper we describe the network infrastructure we designed, built and operated for the International Mathematics Olympiad in the Netherlands in 2011. The infrastructure was pragmatically designed around OpenVPN tunnels in a star topology between the various venues. VLANs were extensively used to separate functional groups and networks. The actual construction of the event network took about 3 days and was needed for only 2 weeks. The architectural, setup, building and operational aspects of the network are described and we include some lessons learned.

1. Introduction

The International Mathematics Olympiad (IMO) was to be held in the Netherlands during a week in the summer of 2011. During this week 600+ scholars plus about the same number of support staff were invited to the Netherlands to do 2 days of mathematics contests. In the weeks before the event. the assignments had to be created and selected by the team captains and jury at an undisclosed place. After the event, all work needed to be scanned, transmitted, corrected and the results and final classification needed to be ready within 2 days.

To support this organization a complex and temporary (as the Olympiad is held in a different country every year) IT infrastructure was needed. At all locations we needed to set up infrastructure in such a way that a complete internal private network for the Math Olympiad was created. We had 4 locations and a central datacenter to be connected for the different services. Owing to the security measures as a functional requirement, we had multiple different networks across the infrastructure. Further, we wanted one single point of access to the Internet for access control.

While all of the networking was needed for about 2 weeks, we only had about 3 days to build the entire setup at the various venues (hotels, sports accommodations) throughout the Netherlands. It was a major technical and logistics undertaking, all done with a group of volunteers. In this Practice and Experience Report we will describe the design and setup of the site interconnection networks.

2. Requirements

The functional requirements for this ad-hoc setup were quite simple: the Mathematics Olympiad needed networking facilities to support the contest in the form of internet access and access to the main logistics and administration system and its backup. Security was defined in terms of access control to different pieces of the data for functional groups of users. Additionally there were a number of “nice to have” features defined, such as video streaming and internet café style setups at various locations. A set of strictly defined technical specifications did not exist.

3. Architecture

We co-located the servers at a datacenter of the masters program of System and Network Engineering at the University of Amsterdam where multiple high speed internet feeds were readily available. In this datacenter we located all central services for the IMO.

The central services, such as the main system that supports all logistics and operations, were located here and were reachable over the Internet for certain functionality such as presentation of results. It was also required that these services be reachable from the internal IMO networks for preparing and selecting the assignments, translating assignments to each of the 60+ official languages, and other tasks. At the different sites, the networks were separated into functional VLANs, which were distributed throughout the different venues using VLAN capable switches.

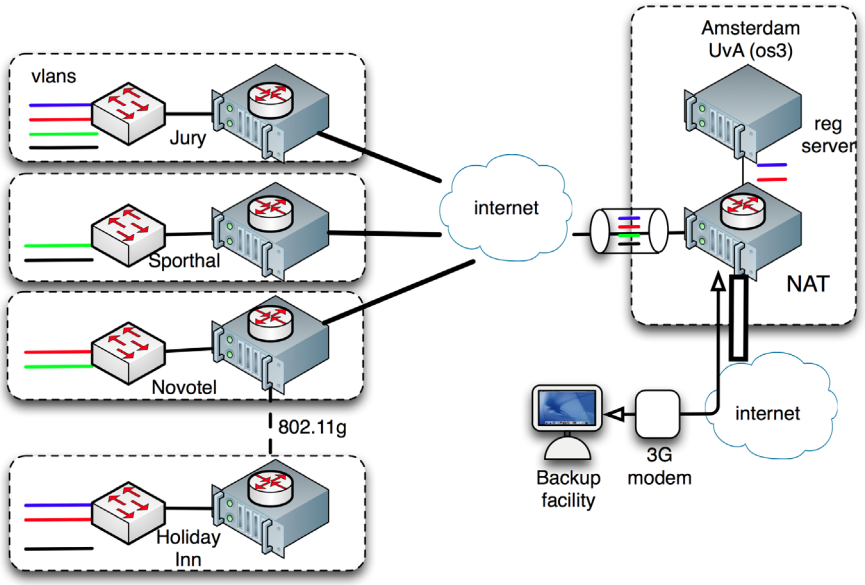

Figure 1: Schematic network overview

All sites were equipped with a VPN tunnel to the central datacenter. All tunnels were terminated at one point. At this termination point we are able to control routing between the different sites and networks. Also, the Internet connectivity was controlled at this point. A schematic overview of the setup is depicted in figure 1.

3.1 VPN setup

To establish the VPN tunnels between the different sites we used OpenVPN. OpenVPN is a SSL VPN solution that works on the application layer, in contrast to other solutions like IPSec where the VPN is created in the network layer of the stack. A specific and important advantage of OpenVPN is that no extra difficulties are caused by the use of network address translation (NAT). We had to cope with very different Internet feeds at the different venues we were using. While NAT traversal was available for IPSEC, the extra configuration complications favored avoiding it entirely. Having had previous bad experiences using IPsec from a system behind a NAT gateway we decided to use OpenVPN. Another advantage of OpenVPN was that it was easy use a non-standard port if needed, to circumvent local firewalling: for example 443 (https) instead of the standard 1194. Because of the fact that OpenVPN runs in user space, very high throughput (> 1Gbit/s) can cause context-switching problems, but the total (VPN) network bandwidth that we would need at any location (including the central site) would not have exceeded 600 Mbit/s in our rough estimations, taking into account the bandwidths we got at the different venues.

3.2 VLANs and IP Number plan

For the internal IMO infrastructure we designed an IP numbering scheme that used RFC 1918 private addresses. We put different functional groups of users in different VLANs: contestants, jury, officials etc. The IP ranges on these VLANs corresponded in one of the octets to the VLAN number to ease identification and troubleshooting, for example VLAN 15 corresponds to subnet 10.15.0.0/12. It proved to be very helpful to identify the network (VLAN) from the IP number issued from DHCP.

3.3 Distribution

All sites had a VPN terminator machine that both served as router and provided for basic network services for the IP ranges on that particular site. The VPN terminators provided network services such as DHCP and DNS (relay) to the local networks. This prevented broadcasts (DHCP) over the VPN and created a fallback scenario; when the VPN tunnel to the central site had problems, Internet connectivity to certain VLANs could be restored easily by using the local Internet connection. The internal network interface on the VPN terminator machine, equipped with a number of virtual interfaces, one for each VLAN, carried a trunk that brought all VLANs to a switching fabric. This setup of switches separated the different VLANs to the users.

4. Implementation

We aimed to implement all network infrastructures as transparently as possible, using mostly Open Source solutions. Since the number of devices in the network infrastructure was rather limited, and the time to build and operate was really short, the decision was made to do a “one-off” and pragmatic design. We refrained from setting up configuration management and a deployment environment, and we defined network monitoring as a nice to have, but not essential to the operation. For example the VPN terminators were “hand configured” in a standard way documented on the ICT project wiki.

4.1 Background

The VPN implementation was done on FreeBSD 8.2. We chose FreeBSD because of a good network stack, easy deployment and a proven OpenVPN port. Each site was equipped with a FreeBSD machine running OpenVPN. We had a hot-spare machine on the central site to be able to cope with disaster during the event. All machines had 2 network interfaces: one connecting to the Internet feed on the location and one on the internal networks (switch fabrics that were installed).

4.2 Central Site

The central site machine was running 4 instances of the OpenVPN daemon, each running on a different port at the external interface. We needed multiple daemons running at once since the OpenVPN daemon was not multithreaded, and to make efficient use of the multicore architecture of the central site. This resulted in a clear configuration; each daemon maintained a distinct connection with a separate configuration file and virtual endpoint interfaces on the machine. This simplified the routing setup and debugging. All VPN tunnels were using self-signed certificates with a private CA for authentication and encryption, installed manually. The bandwidth needs were not so large that we needed to take special measures here. On the central site we had access to 1 Gbit/s full duplex, where on the other sites bandwidths varied between 40 Mbit/s and 100 Mbit/s.

The central machine also provided the Internet connectivity. To connect to public external addresses from the internal provided network ranges we used NAT on the central node. Since we had an estimated 400 international people concurrently using the network, many of whom might be using tools such as Skype, or browsing the Internet, we needed to cope with a large number of outgoing sessions. This discouraged the use of user-space `natd`(8), so we used `ipf` and `ipnat`(5) providing an in-kernel solution that was able to handle sessions more efficiently. This also provided the ability to add multiple external IP addresses to be used as source IPs for the NAT sessions so that we would not run out of source port numbers.

4.3 Remote sites

All remote sites had the same basic FreeBSD setup, where just the OpenVPN configuration files, certificates, DHCP, DNS and routing configurations differed. Also the network setup was kept as simple as possible to prevent errors. When we started the VPN, the endpoint machines were not reachable over the Internet anymore, but only through the VPN tunnel from within the IMO network cloud. So, to prevent lockouts, an extra firewall / routing entry was added on each remote site to be reachable over the Internet without using the VPN tunnel. On the remote sites the internal interface of the VPN terminator carried a trunk with a number of VLANs that were split using switches.

4.4 Switches

To have full control over the network environment, the switches that we deployed were all managed by the IMO IT team. To prevent issues we decided not use the existing switches that the venues had in place. This reduced the number of potential building and operational problems for preexisting, disparate switches enormously, trading them for procurement, configuration, and maintenance of new switches. This reduced the number of potential problems (building and operational) enormously, but faced us with extra tasks to have switches available, configure them and maintain them. The only infrastructure we used that was provided by the venues was the Internet uplink and the dry copper (and dark fiber) to the rooms that they had installed. Use of the existing infrastructure, even just the cabling, proved to be challenging because of the way it was installed (see Figure 2).

Most end user equipment was connected to wired Ethernet. At some locations (local) WiFi was used to extend the wired infrastructure. In order to not run into extra complexity, we deployed simple (unmanaged) access points connected to the wired network when needed. We thought of reflashing the APs with an Open Source distribution, but as we used them simply unmanaged it did not add any value, so we used them with the manufacturer`s firmware

5. Deployment

As a team of about 6 people partly in their spare time, installing the complete infrastructure was a major logistics operation. Key here was that we had ample space in a datacenter of the University of Amsterdam which allowed us to have all VPN terminator machines (central and all remote sites) set up in a lab environment before actually shipping them to the sites.

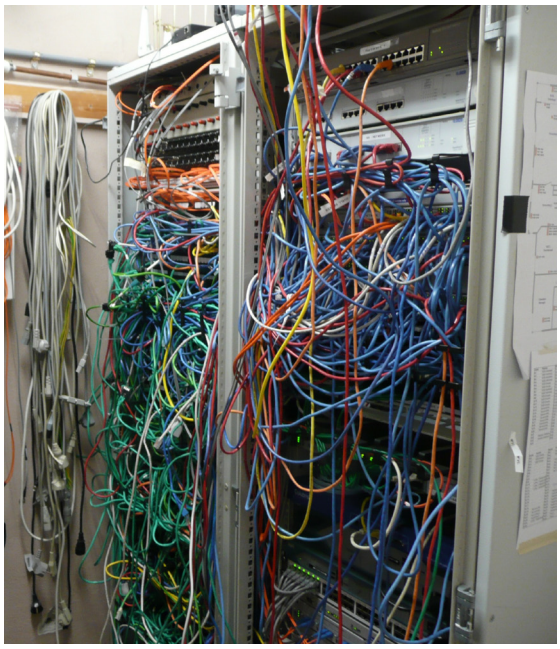

Figure 2: Existing Cabling

Figure 3: Testing in the data center

On an afternoon, about a week before the event, we built the complete server and switching environment in the datacenter to test all configurations and routing. This included a wireless link we were to use between two buildings.. The relatively small setup (5 locations) allowed us to do this just before shipment of the equipment to the remote locations. This exercise proved to be a major advantage, because during these tests a number of configuration issues showed up such as mistakes in IP numbers and configuration errors in the wireless bridge Figure 3 shows one of the authors during the test of the complete setup with all VPN endpoints and switches.

A number of site-surveys on the different locations were held well in advance in order to deploy the network on various sites efficiently. Recognizing the local situation, patch layouts and uplink possibilities proved to be very helpful in the deployment. The site survey revealed that using a wireless bridge between two sites was a better solution than adding another VPN site. Also explaining the design, the setup and the work we needed to do on-site to the local tech staff and contractors created a good working relation during the event.

Another important point in building an event-network setup is that labeling and having / keeping inventory lists (and up-to-date DNS) are critical. The necessity of fast, ad-hoc setup for a short term goal lends to the engineers' ability to improvise and adapt in a fluid environment. But without proper labeling (see Figure 2) and documentation, debugging would have been a serious problem and could have had negative repercussions for the IMO event.

6. Operation

The complete infrastructure was set up in the different venues over 4 days. There were quite a few network layout and connectivity problems that were unforeseen, including physical layer Internet feed issues at one location; these problems required some improvisation and quick thinking.

Other issues that we faced were partly on the political level, and being able to cope with the improvisation talents of other parts of the organization, as rooms and network layouts were changing unexpectedly and without advance notice. As we had configured a number of extra ports in all VLANs on the different switches, and have them clearly labeled as such, it was not a big problem to cope with these ad-hoc changes.

Next to good documentation and a solid communication infrastructure (a wiki and cell phones with all numbers of the organization team at large preprogrammed) good interpersonal communication skills and true dedication within the team were absolutely critical, here. Simple tools like network analysis software (such as wireshark), cable testers and tracers have proven to be very useful in the appropriate hands. Also a labeling machine has proven to be very useful, as can be imagined.

7. Backup

Fortunately, all the measures that we took to prevent and overcome network problems were not needed. We did have 3G cellular modems at two important sites that would have been able to serve as a backup link if necessary. As already mentioned, the central site had a standby machine to be used as VPN terminator, multiple Internet feeds, etc. In case of an emergency on any of the remote sites this warm-standby machine could be converted into any of the VPN endpoint machines on the fly by manual switching configuration directories.

For the main logistics and operational system we had a backup in another data center, reachable over the Internet and a live copy of all the operational data on a laptop.

8. Conclusion

A good and small team, with defined skills, clearly expressed responsibilities and a vast amount of improvisational talent will get you an interesting and high performance multi-site network for an event. However, do not underestimate the amount of time needed for preparation and design. We started this project more than 1 year before the actual deadline of the event. Although the requirements for the network were not strictly defined, a common sense engineering approach and a number of definition and engineering meetings resulted in a very workable, not overcomplicated setup.

The pragmatic approach worked here. But, if the event had been larger or longer running, appropriate best-practices, such as the use of a configuration management system and extensive monitoring systems, would have had to be evaluated.

9. Acknowledgements

The authors want to gratefully thank the masters program for System and Network Engineering at the University of Amsterdam, the rest of the IT team and organizers of the IMO and all sponsors for their help, support and good company. Thanks go also to the authors' employers (Competa IT B.V. and the University of Amsterdam) that supported this work by allowing the time to do this project.