02 – Content Switching

Hello, and welcome to the "Cisco on Cisco" seminar on Cisco IT Content Switching Deployment. I'm Kevin Rochowski, and I'm an IT Project Manager in the IT Cisco on Cisco team. The theme of our show today is content switching at Cisco and how we deployed it for the Internet portal. You'll get a chance to learn what content switching is, why Cisco IT decided to deploy content switching, what the architecture comprises, as well as some of the lessons we've learned along the way. It's my pleasure to introduce the guest of today's show, Koen Denecker. Koen is an IT engineer based in Amsterdam. He has been with Cisco IT for almost five years now and is the Content and Application Networking Domain architect in the IT infrastructure department. Koen, thank you for coming today. Thank you Kevin. It's a real pleasure to be here and to be able to present the best practices and lessons that Cisco IT has learned about content switching. So sit back and enjoy as we explore how Cisco IT is using content switching to support our web presence strategy, and provide the highest levels of URL virtualization and availability.

OUTLINE

So let's start with an outline of today's presentation. I'd like to start with giving you an overview of what content switching is all about. Then we'll look at the server switch architecture which is how IT deployed the content switches in our infrastructure. We then move into a case study and that's going to be about our internet presence on www.cisco.com. We then widen our horizon and look at the wired content landscape to finish with a second case study which is going to be how we virtualized the e-mail status internally. We then end with a question and answer session.

CONTENT SWITCHING OVERVIEW

So let's start now with giving you an overview of what content switching is all about. So what can content switching do for us? To do that we'll start with a number of terms I'd like to introduce.

CONTENT SWITCHING 2

And the first one is a server farm. By server farm we mean a group of one or more servers with nearly identical configuration and providing equal functionality. You can think of a barrel of DNS servers or a barrel of web servers. The second term I'd like to introduce is server load balancing. By server load balancing we mean the distribution of incoming data connections across servers in a single server farm such that the overall capacity and availability increase. Server load balancing typically takes place at either Layer 3 or Layer 4 and is also introduced as the concept of a virtual IP address. This one is typically hosted on the server load balancer. The third term is content switching. Content switching is all about the intelligent distribution of those data requests across servers which may now be in multiple server farms. And the server farms may have different functionality. The goal is to increase the overall capacity, availability and/or functionality. The bottom right shows you an example of that. We have two server farms. One which is a web server farm for static content, and a second one, the orange one, which is to serve dynamic content. The virtual IP address is hosted on the content switch, and an incoming HTTP request is parsed to decide whether it should go to the server farm serving static content, for example a GIF image, or serving dynamic content.

CONTENT SWITCHING 3

Now let's ask ourselves why do we need content switching, and there are a couple of reasons for that. The first and foremost is high availability. High availability is a business requirement which we see in all mission critical application environments. Content switching can provide high availability through a number of mechanisms. It can do so through load balancing, but also through health monitoring. So if a server breaks it's taken out of the farm. It also allows for proactive server maintenance. If we need to work on a server we can take it

out of service and put it back in line after the fix. The second reason for using content switching is functionality. In an ideal world the front end, so the user facing web infrastructure should be separated from the backend. The front end should be user friendly and remain unchanged. Whereas for the backend we want high levels of flexibility to provide additional functionality as we move along. For example, we're going to talk a little bit in more detail about what we call retaining the www.cisco.com paradigm. A third reason for looking at content switching is to do cache control. Often, although the content engines have default cache logic, we may want to have more granular and flexible control of what is being cached and not. The fourth reason for doing content switching is security. First of all by terminating requests on a virtual server we provide some hiding of the backend and we can also provide some levels of protection against denial of service attacks. Secondly, the content switching is enabling the ease of technology to support SSL offloads through the use of the SSL service module.

THE WWW.CISCO.COM PARADIGM 1

Let's have a closer look at what we mean with the www.cisco.com paradigm. And as always, as I already mentioned, it's all about separating the front end, the URLs the user are facing, from the backend which is designed for flexibility and scalability. The diagram here shows how a user makes a request and through DNS the domain name gets resolved to a virtual IP address which hosts on a content switching module. Then a first rule is applied which to look whether the content is cacheable or not. And if it is it is sent to an ACNS content engine farm. But then additional rules come into place. So we really deploy a technique called URL switching here. As you can see there are multiple server farms. One is designed for software distribution, static content, other ones may map to multiple application environments. As you can see the user side is virtualized into a virtual service irrespective to your graphical location or actual content being requested.

THE WWW.CISCO.COM PARADIGM 2

Let's have a closer look at what we mean with the www.cisco.com paradigm. And as always, as I already mentioned, it's all about separating the front end, the URLs the user are facing, from the backend which is designed for flexibility and scalability. The diagram here shows how a user makes a request and through DNS the domain name gets resolved to a virtual IP address which hosts on a content switching module. Then a first rule is applied which to look whether the content is cacheable or not. And if it is it is sent to an ACNS content engine farm. But then additional rules come into place. So we really deploy a technique called URL switching here. As you can see there are multiple server farms. One is designed for software distribution, static content, other ones may map to multiple application environments. As you can see the user side is virtualized into a virtual service irrespective to your graphical location or actual content being requested.

SERVICE SWITCH ARCHITECTURE

Now that we talked about what content switching is all about let's have a closer look at the server switch architecture, which is the way Cisco IT has architected and designed content switching as a service in all our data centers worldwide.

PRODUCT EVOLUTION

To do that we have, over time, used a list of various types of hardware. To start with, on the left, you see the oldest model. That was the local director 430. In the middle you see the content server switch, the CSS 11503. And on the right you see the content switching module, which is a Catalyst 6500 module. The local director had a fairly large scale deployment in Cisco IT. It does suffer from a number of drawbacks though. First of all, one local director pair is required for each server farm. In other words, it cannot server any server farms in more than one VLAN. In addition to that is it's only a load balancer. It only operates up to Layer 4 and has little or no capabilities to do intelligent content switching. We had a fairly large deployment but we're moving out of that product. The second product that came along, the CSS 11503 had a limited adoption. It shared a drawback that there was a requirement to have a pair in each VLAN. But the good thing about the CSS was that it was the first true content switch. So it did offer us full Layer 4 to 7 capabilities, which is something we had a

strong business requirement for. And then on the right you see the strategic platform. So Cisco IT has decided to deploy the CSM in all our datacenters. And it's great for a number of reasons. First of all it fully integrates into the network, not only at Layer 2 and 3, so with VLANs and subnets, but also it provides us unmatched Layer 4 to 7 capabilities and high scalability. In addition to that it's fully integrated into IOS, which was much appreciated by the support staff.

ARCHITECTURAL REQUIREMENTS

Before we provide the architecture let's have the requirements, which drive the architecture. The first one is all about cost and value. Cost should be lowered where possible and when possible. And this is achieved in the following way. Whereas in the old world we used to have a dedicated pair for each server farm or each VLAN, in the new world with the CSM – because it supports multiple VLANs and subnets – you need only one pair to serve an entire datacenter. The initial cost for the product may be higher, but in the end the overall cost of all products pulled together and the overall capacity we get for it is lower. The second requirement is around flexibility. It should be an enabler for all other content related services we want to deploy. And those include caching, application acceleration, security, and maybe other functions and services in the future. In addition to that we did not want to redesign or re-architect the datacenter purely for the sake of providing content switching services. So as you'll see then later in the design, we were able to keep our existing subnet and VLAN design. From a security perspective it is also important that it does not introduce any new risks, and it should be an enabler for the other technologies we were looking at in the security area. In addition to that, by introducing a new platform we want access control and denial of service protection on the virtual status. The device itself should be managed in a secure way. A fourth requirement is around scalability. The numbers we were looking at are more than one Gigabit per second throughput. And in addition to that between 2,000 and 10,000 new Layer 7 connections per second. And that really is dependent on the kind of data center where we are targeting. Preferably this should all be hardware assisted such that the software and the CPU can focus on the overall management of the device. Another requirement is high availability. Introducing this should not introduce any new single point of failures. In the target SLA we're looking at is 99.995% availability. Moving forward we want to further raise this to 99.999%. The final point is manageability. Looking at the content switch, and any time in general, when we introduce a new piece of hardware we're always looking at using the existing skills set, the existing management interfaces and tools if we can. And fortunately we were able to do that with the content switching module. In addition to that we want to design it in such a way that our troubleshooting capabilities are not impacted in any negative way. And one specific requirement we had here from the application team was that all client IP addresses should be logged on the server and notified.

SERVICE SWITCH ARCHITECTURE 1

Now let's start and build up the services architecture. What you see here is a standard datacenter design. The upper half shows you the Layer 3 distribution or the aggregation. We call it the datacenter gateway layer. In the bottom half you see the access layer. The goal of the access layer is to provide physical connectivity at Layer 2 due to real servers.

SERVICE SWITCH ARCHITECTURE 2

To provide content switching services through the content switching module we first looked at the option of putting the CSM module directly into datacenter gateways. However, since there was an IOS version dependency we decided not to go down that path. But only for the dependency of the CSM IOS but maybe for IOS dependencies on other modules. Therefore we introduced a new layer, which we call the server switch layer. Server switch layer consists of two server switch chassis, which have dual uplinks to the datacenter gateways.

SERVICE SWITCH ARCHITECTURE 3

The server switch itself is a Catalyst 6500 or the 7600 chassis with multiple modules. As you can see on the bottom left we have one or more Supervisors, one or more uplink modules, one or more CSM modules, and one or more other modules. Other module may include the SSL module, AL module, or network analysis module. As you can also see, we have a CSM module on each server switch that they operate in an active standby way. So only one CSM will forward traffic or bridge traffic at any given point in time.

BRIDGED MODE CONTENT SWITCHING

Now that we looked at the architecture let's move forward into the design. The design we opted for is called bridge mode. What we do here is we implement a CSM; we put it in the data path in a bridging mode. What that means is that at Layer 2 a new VLAN is introduced and the real servers are put in the new VLAN. From a high availability perspective only one CSM module is active at any time. This is illustrated in the three diagrams you see on the bottom. On the left you see the Layer 3 view or the IP addressing view. There are no changes here to the existing real server IP addresses. The only thing we do is we add a new virtual IP address in the subnet. That virtual IP address is hosted on the CSM. Then on the middle you can see the Layer 2 view. And this is where it all happens. That's also why it's called the bridge node. Before we introduce a CSM the real servers were put into the left VLAN, the VLAN 301. But with the introduction of the CSM we add a new VLAN, 601, which pairs with the 301 VLAN. In the real servers VLAN assignment is being changed to the new VLAN. And as you can see the only way to get from the datacenter gateway to a real server is by bridging through the CSM module. This gives the CSM the opportunity to operate on the packets and rewrites IP addresses, MAC addresses, or operate at the higher layers of the model. The bottom right shows you then the physical view. The physical view is identical to the previous slide. And here you can see how the CSM physically resides in the server switch. There are advantages and disadvantages to this design. There's no perfect design. Let's have a look at the advantages. The biggest advantage to us is that there are no Layer 3 changes to the datacenter; there are no changes to the service. Another advantage is the ability to segregate the VLANs. We introduce a new pair, 301, 601, and if you introduce another pair, 302, 602, then we can set it up such a way that there's no traffic leaking from 301 to 602. If we look then at the disadvantages the major disadvantage is due to the fact that all traffic must now physically flow through the CSM. This means that backup traffic but also non-virtual server IP traffic needs to flow through it. This may be a capacity issue, but the workaround we looked at there was to put all the backup traffic through a dedicated interface on the real servers. A second minor risk is there is an operational risk of introducing spanning tree roots. This shouldn't happen by experienced engineers. The final disadvantage is about the troubleshooting. Because everything operates in hardware it's not trivial to analyze and inspect a specific session. The workaround we employed for that is we also used a network analysis module, which is being deployed in every server switch.

IT ADOPTION STATUS HARDWARE

This slide now shows you the IT hardware adoption status of the various hardware types over time. A long time ago we only had the local directors, and we had in excess of 50 pairs to serve all our needs worldwide. Then came the content services switch, the CSS. Because it gave us the Layer 4 to 7 capabilities we had a ramping up deployment of about 30 pairs. Where we are now is that we introduced a CSM into our infrastructure and that is taking over the functionality from the local directors and the CSSs. So the number of local directors and CSSs has gone down already significantly. But moving forward the target state is that we would only have CSMs. As you can see in the target state there's no local director, no CSS, and we have about 25 CSM pairs worldwide and those are residing in 10 to 15 service grid sessions. So it wasn't for one for one hardware replacement? No Kevin, it wasn't.

IT ADOPTION STATUS SERVICES

As a matter of fact we went from an old world where we had one pair of hardware per VLAN to a new world where we had one pair of hardware per datacenter. But on top of that we were able to significantly increase the number of services. As you can see in the old world we had about 100 virtual servers on 50 pairs. In the new world we have more than 600 of them, and that is only on 25 CSM pairs. The key value here to us is agility. In the old world, when we wanted to deploy a new virtual server it also meant we had to deploy new hardware. But in the new world we are now able to deploy a new virtual server with no new infrastructure requirements. And that only takes us a couple of days. And we are hosting some of the most critical applications on the CSM. I've given a couple of examples here. www.cisco.com that serves more than 50 million Layer 7 decisions on a daily basis, and it's going to be the topic of our case study. Another record is set by our instant messaging solution running same time. Here we see peaks of more than 20,000 simultaneous connections. A very business critical application environment is the Oracle 11i manufacturing environment. Another one is Livelink for collaboration. We

also deploy external and internal Java 2 Enterprise Edition applications behind the CSM. And finally we use it for the Microsoft Exchange front end. Now let's move into our case study.

CASE STUDY 1 WWW.CISCO.COM INFRASTRUCTURE

The case study is all going to be about how we deployed the content switching module for www.cisco.com.

CISCO CONNECTION ONLINE (CCO)

Internally we use a word CCO. CCO stands for Cisco Connection Online, and it represents Cisco's public web presence. It is an infrastructure, which is responsible for more than 90% of Cisco's revenue. On average this equals about $2.5 million dollars on an hourly basis or $42,000 per minute. And this is just an average. In peak times this can be five to ten times higher. And as you can imagine, any outage in this environment means an almost immediate loss in business and is very critical.

OVERVIEW OF CCO MIGRATION

The CCO migration is part of a larger roadmap. And when we talk about CCO migration we mean the movement of a local director based environment to a CSM based environment. But that wasn't just to increase scalability and availability. We also wanted to increase the flexibility. And in doing so it enables us to support application specific architectures. The side effect would be that the total cost of ownership of the infrastructure should go down.

OLD ENVIRONMENT LOCAL DIRECTOR

Lets first look at the old environment. This is shown in the diagram on the left. On the top you can see how a customer or a client would initiate a web request to www.cisco.com. It would first hit a WCCP enabled router. WCCP is a web cache communication protocol that will intercept the HTTP request and send it to the cache farm. The cache farm will analyze the request and serve the request from the cache if it can do so. If not, for example because the content is stale or because authorization is required, then it will send it to the virtual server residing on the local director. The local director will then pick a server in the front layer. The front layer has a number of web servers, but we are talking about load balancing here. So as you can see there's only one server farm. The server farm will respond and may go back to the additional layer in a classical multi-tiered web architecture design.

NEW ENVIRONMENT CSM

And we go now to the new environment, which is, CSM enabled and we see the following. First of all the virtual server will now reside on the CSM and we will no longer use WCCP for intercepting a request. The CSM and the virtual server have rules, and the first rule we're going to hit is a caching rule. A caching rule is going to say is this content type cacheable or not? And if it is it's going to be sent to the ACNS content farm. If not it will pick an appropriate server farm, and this is where URL switching comes into action. As you can see we now have multiple server farms in the front end. One, for example, could be running on Solaris, another one could be running on Linux or Windows. And they may have different functionality as well. And then from the front layer we move to the back layer and to the backend databases. So the key differences from the old environment to the new environment are as follows. First of all only cacheable content will flow through the ACNS reducing the overall throughput requirements on the content farm. Secondly the overall capacity and scalability will go up. And thirdly the flexibility will improve because we can now support multiple server farms and we can migrate applications from one application environment or server environment to another one.

MIGRATION IN MULTIPLE STAGES

The migration of course didn't happen overnight. It actually took us multiple stages and it involved both infrastructure and application themes. The first step of course was to deploy the hardware into the chassis. Then the first official step was that the development environment on Solaris was put behind the CSM. This was done by the infrastructure team and gave the application teams an opportunity

to test their development environments. In a second step we added a second server farm. This time it was running on Linux. Again, the development team had now a chance to test a new application running on the Linux environment. At this stage we do support content switching in the development environment. Then step three is about doing exactly the same thing in a stage environment. Again, first the infrastructure team takes the actions on configuring the CSMs, then the application teams verify that it is all working as expected./l”{jhujikol,mkjolkhyyyorekw0pies0ir0saer0et,s09y4,ye Now we're getting ready to do the big production change. But this itself was not one step probably but two steps as well. Initially we introduced a new fully qualified domain name which was only available internally. We could call that a pre-production environment. That gave the application teams an opportunity to test. And at this time we had very stringent QA requirements. And then we were ready for the big go live. On a regular maintenance window on a Saturday evening the infrastructure team did the flip over, and at that time the application team was ready to verify that all applications were working fine. So, how did it go? Well Kevin, to be honest, it didn't go as good as we hoped it would.

FIRST ATTEMPT GO-LIVE ISSUES

On the first attempt we had a couple of issues with going live. What happened was, as we enabled the new virtual server on the same virtual IP address we had an outage which lasted for a couple of minutes. And we didn't really immediately know what was going wrong. When we started troubleshooting we saw that a local director did not relinquish the MAC address of the virtual. Because it's such a critical piece of infrastructure we had to back out of the change. We then went into the lab, we were able to duplicate the issue, and did it again. We found the workaround, which was essentially just restarting the local directors as soon as the new virtual, is live. And when we tried it a second time it went fine without any issue.

WHAT NEXT?

The overall program continues of course. After we did the virtualization we now have an environment where we are where we have increased levels of flexibility on the backend. As we introduced a new Linux environment we will now also upgrade the Solaris environment and add new services such as eCommerce and FTP to move them behind the CSM. In parallel with that we are looking at deploying SSL, or secure socket layer, to encrypting logins and the payload.

CCO SSW PERFORMANCE

In the meanwhile the operational teams have supported the infrastructure, and this is the information they wanted to share. What you see here is the throughput on each of the uplinks of the server switch. This is over a two week period, and you can clearly see the day of the week patterns. The dashed line represents 100 Megabit per second throughput. And you can see if we add these together, that we regularly exceed 200 Megabit per second. This is fine since the CSM supports up to 2 Gigabit per second. So we are well within the capacity provided by the CSM. In addition to that, if you look at the number of new Layer 7 connections, so the number of URL switches that the CSM is doing, that we see that on average we have about 500 of them per second. However, in peak times we saw up to 10,000 of them, and that may include security attacks. Again, this is well within the capacity of the CSM because it can support up to 70,000 of them.

LESSONS LEARNED

The lessons we learned here are the following. First of all from a manageability perspective the CSM is a great product because it allowed the operational engineers to use a well known CLI and a well known existing management and troubleshooting tools. This is a big difference from the CSS where we found that introducing it took quite a number of months to get the engineers up to speed. The second point around manageability is that not just the networking teams are involved, but now we also involve in the application teams. Through the use of TACACS we are able to delegate access to other people even though the networking teams own the hardware. From a caching control perspective, through the use of the Layer 7 rules, instead of WCCP we have a much higher level of web cache control. The first specific issue was around Legacy applications. We found that we had a number of applications which were more than five years old and didn't deal well with the caching logic. Through the CSM you can very easily bypass them. The second one is that the load balancing is

much more sophisticated than what WCCP allows us to do. A third one is that the Layer 5 rules, which are growing over time, are processed in hardware. If you do this on the content engines then they're done in software. So over time as they grow you may see the real performance going down, an issue we're not seeing on the CSM. And since only cacheable content is now sent through the cache farm, the overall load on the ACNS content farm is being reduced. Another lesson we learned a couple of months after the go-live was that the default URL parsing link, which is only about 4,000 bytes, was insufficient. Some applications use very long cookies. But the good news here was that the CSM allows us very easily to increase that default parsing link.

CONTENT LANDSCAPE

Now let's widen our horizon and look at the bigger landscape within Cisco IT.

SSL SERVICE MODULE

The first one I want to drill down a bit more in detail is about the SSL service module. The SSL service module does SSL termination, both encryption and decryption in hardware and it is also a Catalyst module. It fits well into service switch architecture, and it is being used in two scenarios shown on the right. The first scenario is all about SSL offload. Here rather than terminating the SSL session on the real server we terminate it on the virtual server. The virtual server, which is hosted on the SSL module, does the SSL decryption and then sends the traffic in clear text to a real server. The reason for doing so is performance without any compromise on security. The second scenario we're looking at, SSL backend re-encryption. It starts similar to the SSL offload, but then we apply content switching to the traffic flow and do the re-encryption. Here the SSL module operates both as a server and as a client. And the key reason for doing so is that allows us to do intelligent content switching on encrypted traffic. Without this the CSM has no visibility into the Layer 5-7 capabilities of the flow. But by opening a tunnel, inspecting the URL, making a decision and closing the tunnel again, we are now able to do so. The benefits of using the SSL service module are multiple. By terminating the SSL a little before the real server we are able to use our payload inspection solution. Second one is about performance. Because everything is done in hardware we are able to achieve much higher levels of capacity. A third one is around functionality. Without the SSL backend re-encryption scenario we are unable to use the content switching paradigm for SSL traffic. And a fourth one is around agility. By using the SSL module we are now able to do provisioning of SSL services much faster than we used to do.

SSL SERVICE MODULE NETWORK DESIGN

The design is similar to the CSM design; however, it's slightly different from an SSL perspective because the SSL modules are now deployed in a routed mode. In other words, the SSL service has its own IP address and its own subnet as well as its own VLAN. This is shown on the three diagrams, which are similar to the CSM diagrams we saw earlier. From a Layer 3 perspective we do add one new subnet for the entire datacenter. That subnet hosts all SSL proxy virtual servers for the entire datacenter. Of course introducing a new subnet also requires a new VLAN. This is shown in the middle diagram and it's called the VLAN 901. As we deploy other content service solutions the single VLAN should be able to handle all of them. And then on the right you see the physical layout. In a physical layout all we do is we expand on the existing server switch paradigm and we deploy an SSL service module in the existing chassis. The adoption, from an adoption perspective, we've deployed both scenarios. In terms of intelligent content switching we have been using this for quite a while now in the Java 2 Enterprise Edition environment where we are able to support multiple backends from different vendors over SSL. Another example of adoption is the front end for Exchange, and that's going to be the topic of our second case study.

SSL INTELLIGENT CONTENT SWITCHING IN J2EE 1

Here's a big more of a drill down of what we do with our Java 2 Enterprise Edition environment. On the left you see the client side and on the right you see two backend application environments to support Java 2 Enterprise Edition. And in our situation we have environments from different vendors. So normally this would be an end-to-end SSL session. But the problem is you really don't want to change your URL when you change the environment from one backend to another one. So what we do is the following. A client initiates a request over

SSL then hits a virtual on the first CSM. Since we talk about encrypted traffic, the content switch can only look at Layer 3 and Layer 4. It picks one of the SSL service modules, the first SSL proxy. That one does the decryption. By decrypting the traffic we now have full visibility into the URLs. So we send it back to the CSM. CSM has a look at the URL being requested and basically maps it to one of the backend environments. Now since the backend environments are expecting SSL we need to encrypt it again. Very simple, we send it back to the SSL module, the SSL module now acts as an SSL client, does the encryption and then sends it to the backend environment. But to do so it hits another CSM again to achieve high availability. And this is how it works logically. Now these are about five steps in the data flow. And do we really need a physical layer for each of those five logical layers? No, we don't.

SSL INTELLIGENT CONTENT SWITCHING IN J2EE 2

This is illustrated in this slide. And as you can see all content switching requirements, whether it's at Layer 3/4 or at Layer 5 to 7, are hosted on the same pair of content switching modules. In addition to that all SSL services, whether we talk about encryption or decryption, are performed on a single pair of SSL service modules. In addition to that these four modules do all the content service as well for the entire datacenter.

CONTENT LANDSCAPE GLOBAL SLB

Moving forward let's also have a look at the bigger content landscape in terms of global server load balancing. So while CSM is a target platform for local server load balancing, we also have a business requirement to globally distribute requests to different geographical areas. We do this both externally and internally. Externally the adoption is fairly limited, and a typical example here is ftp.cisco.com. However internally we have quite a bit of mix and match of technologies to meet a large range of business requirements. The table shows the list of products we are using internally, and as you can see each of those products uses different methods, has its own advantage and disadvantage. Also note that two of them are Cisco IT developed solutions. That was done either because no product was able to meet our requirements or to fulfill a specific historical niche requirement. The first product is the distributed director. This is our target platform for globally distributing server requests. It uses DNS as its core mechanism, and it integrates with the network through the DRP, Director Response Protocol. The biggest advantage is the robustness of the solution, and OP1 solutions are deployed on distributed directors. The disadvantage though has to do with the limitations of DNS. The granularity is limited to your nearest DNS proxy. In addition to that you are subject to the caching control in DNS. A second solution is ESL, which stands for Enterprise Service Locator. It is an IT developed solution, which uses the network topology at an active directory site granularity to achieve higher levels of granularity. The third one is the ACNS content router simplified hybrid routing solution. This is nothing else than a static routing table to map certain requests to certain servers. It uses the concept of a coverage zone, and the biggest advantage is the control. By statically using these mapping tables we can achieve very high levels of control. The big challenge, of course, the management overhead. Another technology is WCCP, Web Cache Communications Protocol, which uses the router intelligence to redirect requests transparently. The advantage here is the lack of management. Another one is the global site selector. This is currently being evaluated by the IT department and is targeted as a follow up target platform for the distributed director. A final one is proximity, and that was developed to meet very specific requirements we were facing in a streaming environment. There we want to control the bandwidth industry as well as the transport mechanism, multicast or unicast, for specific lines.

CONTENT LANDSCAPE ACNS

The final topic in the content landscape is ACNS or application and content networking systems. Within Cisco IT we have deployed ACNS globally to every office. And we've done so in a tiered architecture. We now have more than 300 content engines, and they are used for various business purposes. The first one is around software distribution. Here we use the content network to distribute software, not just applications but also entire operating system images. The second area is streaming media. All our video on demands as well as all live streaming is supporting using a combination of ACNS for on demand and live splitting, and the IPTV product. A third area is security. We can leverage the URL and proxy capabilities of the content engines to provide full URL filtering. And we do so at the Internet edge, at the

WAN edge, and at the lab edge. The final business call is to accelerate web applications through caching. What we see is that many web-based applications have a large number of small objects and these are often cacheable. By caching these on site we can significantly decrease the WAN load as well as improve the user interaction.

CASE STUDY 2 – EMAIL SERVICE VIRTUALIZATION

This brings us to the final case study, which is all about virtualizing the e-mail service.

EMAIL SERVICES

This is going to be our relying on the Exchange platform to provide e-mail. And worldwide we have about five datacenters which host a centralized exchange deployment. The requirement to virtualize e-mail was to do with SMTP, POP3, and IMAP4 and HTTP access to those Exchange mail service. In doing so we didn't want the end users to point to specific physical servers. Rather we want them to point to one well known fully qualified domain name. In our case, mail.cisco.com. In parallel with that we had some security requirements. The security requirements are all about not having to send any credentials in clear text over our network. So for HTTP we want to enforce SSL. For POP3 and IMAP4 we want to make it available. Not enforcement because we have a number of Legacy scripts which rely on POP3. Since SMTP does not send the user name and password there was no requirement to do SSL encryption here.

SOLUTION STEP 1 LOCAL SLB

So let me take you step by step in building this solution. So we start with the Exchange environment. So we have a redundant pair of servers in, let's say, five locations distributed over the world. Here we add the content switching module to provide a virtual server. That gives us local server load balancing and high availability.

SOLUTION STEP 2 DNS-BASED GLOBAL SLB

Then in a second step we add DNS based global server load balancing. Here we can use the distributed director or the global site selector to introduce the fully qualified domain name mail.cisco.com, and assign it to all the virtual IP addresses distributed over the world.

SOLUTION STEP 3 ADD SSL OFFLOAD

Now the third step we add the SSL offload capabilities. We have the SSL service modules in every datacenter already. And all we do is add SSL proxies to that. This is then how we achieve the ultimate goal of end users only having to know the fully qualified domain name and being able to securely connect.

END RESULT

In this example the user opens the browser and there's mail and automatically it is being redirected to a secure environment. He has to authenticate, he can access his e-mail and securely. Please know that we were able to do this without deploying any new hardware. All we had to do is verify in the lab that it worked end to end and configure everything in software. And this concludes my presentation.

Q AND A

Thank you Koen. We've put together some questions that customers have asked us about this topic in previous sessions. We have a little time left, and if it's okay with you, Koen I'd like to present those questions to you.

Q. First of all, the service modules integrate at both the network and application layers. These layers are owned by two different teams in my organization, the network team and the application team. How did Cisco IT address this issue?

A. Well, that's a very good operational question, Kevin, and it is one we did face internally as well. In the old world when we only had load balancing it all stopped at Layer 4. And it's okay to ask a network engineer to support up to Layer 4. Now with the Layer 5 to 7 capabilities,

indeed, you introduce a new team in the network area. And they were not really up to speed on that. So really what we did is two things. Short term we brought them up to speed in operating the network equipment, and delegated access to a couple of application engineers who were able to support the environment. That was sufficient to go live. In the meanwhile what we're doing is we are using the capabilities, which are available through TACACS+ to delegate and authorize specific access to their environment, to those application engineers who need it. So we're really drawing a parallel with the server environment. In a server environment the server team knows the hardware and the operating system, and then the application teams know the application and the configuration of it. And we see this taking place as well in the networking area. Okay, thanks.

Q. Secondly, the CSM configuration is integrated into the IOS CLI. Now this includes all rules and policies up to Layer 7. How does this scale for an enterprise with hundreds of applications?

A. Again, another good question. So internally the number of applications we were facing is 2,500 applications, both external and internal, external specifically. So in the CCO environment it's about 800 applications. That's a large number of applications. And then you have to decide very early on in your architecture, in your design, how far you want to go. And there are multiple levels. At the most detailed level you can allow application teams to configure Layer 5 to Layer 7 rules for each application individually. This clearly doesn't scale. The number of application owners is huge, we do not want to delegate access, and the configuration would grow too large. The second one is that you would have rules. Like one rule for one application. So application one goes there, application two goes there, etcetera. That would be acceptable but it wouldn't give us room for future growth. We'd still look at 800 policies. So the thing that IT has done is we went even one layer higher in the options. And what we did was we grouped applications according to the backend infrastructure requirement. So when you look at those 800 applications what we really have is three or four backends. There's a caching backend, there may be a Solaris backend, a Windows backend, a Linux backend, and multiple Java 2 Enterprise Edition environment backends. And that's the level of granularity you want to go. As a request comes in we have huge URL maps that basically say, well, these URLs go to place A, others go to place B. That's the level of detail we allow. The overall configuration is acceptable. And at the same time we are still able to reap all the benefits associated with being able to do URL switching. Okay, thank you.

Q. We're moving towards a service-orientated architecture, which includes virtualization at the server level. Can the CSM deal with this?

A. The short answer is yes, but there's a bit more to it. Within IT, like many customers, we are also moving to a service-oriented datacenter. And that includes virtualization of the service. So multiple soft servers run on a single hard server. Now the CSM can deal with that very well because it's bridging. So you really have to look at the place in the network. If this is the CSM, this is your virtual server and this is your client, then you flow through the CSM so it can do anything it has to do on the packets. The tricky part is that you now have multiple virtual servers in that backend, but those virtual servers only have one physical connection into your network. Going back to the multiple VLANs, your virtual farm is in the front or in the back. And if it's in the back the challenge you need to focus on now is what happens if there's a client on one virtual server needing a service on another virtual server. And that's the challenging part. And by default networking will say, you just go straight there. If you go straight there you are not being bridged through the CSM. So you cannot do the Layer 5-7 content switching. The workaround for that is you have to translate the source client address. So in other words, the client IP address is being NAT'd, and that's the way we pull traffic back into the CSM and then map it to the real server. It scales reasonably well, but it's something you have to incorporate into your design to be able to fully benefit from it. Okay, thank you.

Q. It seems necessary for IT to support quite a number of parallel solutions in order to support the requirements for global server load balancing. Now you mentioned a distributed director, the global site selector, ACNS simplified hybrid routing, and some IT developed tools. Do you think you could maybe explain a little more about this?

A. Yes, that's a good observation there. It is true we do have more solutions than we'd like to do. And that is really the result of having conflicting business requirements on the one side, and capabilities on the other side. If we look at the requirements what we see is that the range of requirements is huge. If it were as simple as all we need to do is get to the nearest datacenter it would be easy. But what we have is we also have functional decisions to be made; like I want the multicast stream or I want a low bandwidth unicast stream. That can't be done

at the DNS layer. And then also some want to go to a datacenter but other solutions need to go as very specific as a small office of maybe only three pieces of equipment and five users. So the range of requirements is really huge in many dimensions. On the other hand we can only do what protocols and standards support. Specifically the DNS standard, it has some functionality to be able to globally distribute requests but it wasn't really designed for it. So you have to make a decision, do you want to go DNS or you want to go application specific. The good thing about DNS, it works for all applications, but if you go application specific you have to really use whatever the protocol allows you to do to meet your requirements. So there is no perfect solution to the problem. That is the historical and the reality of what we are facing today. The key message I'd like you to take away from here is that for the bulk of our applications you need a combination of global server load balancing at the rough level. So you go to US West Coast, East Coast, and Europe. And then within a datacenter, because DNS cannot failover in a matter of seconds, you need the high availability as well, including content switching using a CSM. So the distributed director or the GSS used for the global server load balancing goes hand in hand with the CSM for the local servers load balancing. And that combination serves pretty much 95% of our requirements. I see that's a very good explanation thanks. We have some more questions still. We have one here.

Q. Do you use the CSM for your streaming architecture?

A. We do to some extent. So our streaming architecture is a combination of a multicast enablement streaming architecture and a unicast streaming architecture. Multicast is used on the LAN and we use it because we can leverage the scalability of multicast to provide high bandwidth video to users on the LAN. At the same time in parallel we have a unicast streaming architecture. The unicast side is used for external users, users of the wireless and users over VPN. Now, high availability in multicast is achieved through the network itself and having a standby source. In a unicast area, though, we are using the ACNS live splitting capabilities together with a highly redundant media hop. Redundancy in a media hop is obtained by using the CSM. So the CSM does what it does for static content as much as it does for dynamic content. If you just use it for streaming content that works fine. And in a redundancy in the network as we move away from the hop is achieved through built-in capabilities in ACNS. In other words, the combination of all these technologies allows us to do both a high/low multicast, unicast streaming solution with high availability for all streams.

Q. And how do you monitor the health of your real servers, and how do you go about upgrading these servers?

A. Okay, so that's a question of a more operational nature. The way we monitor -- so the first part of the question is all about how do we monitor and detect that a server is overloaded or broken. The CSM has built-in capabilities for that. It uses probes. That's the terminology being used here, and it has default probes. Default probes are TCP handshake or a UDP packet, send packet, return a packet. Or even at the HTTP layer you can send a certain URL and say I expect this response code. That meets, again, 90%, 95% of our requirements. We do have some advanced requirements now. Specifically the example I want to call out here was the MeetingPlace architecture. In our MeetingPlace architecture we have web service and audio service. And what we want to do is if we don't want anyone to connect to the web server is the audio server is not operational. So you might say you want to implement a blade-sharing. If the one goes down the other one should go down as well. So it's more advanced logic in terms of probing. So the way we do this is we have written a TCL or a TCL script. And that script basically implements the logic we want to see behind the probe. So that works very well as well. So the capability to using TCL scripts is the way around that. And another example I want to call out here is we, at one time -- so normally the CSM will do a TCP handshake. But we had an old server that didn't allow you to reset TCP reset at the end of the handshake. And so what we did is we wrote a custom script, again, a TCL script, and it uses the TCP FIN flag instead of the reset flag. And that was another way to get around it. And that sort of answered the first part of the question, so how do you probe, monitor and take an action.

Q. As for the second part of the question, how do you go about maintaining your servers?

A. Here we have the capability for taking servers in-service and out of service administratively. So a typical scenario would see two servers and there is security identified in the operating system, we need to patch them. Patching may require a reboot. What we don't want to do is just go in there and patch and wait until the probe says server down. What we really want to do is, and what we do is we go in there in the CSM during our change we say that physical server; we take it out of service. So for a short amount of time all requests go to one or two

but not the one we want to work on. We work on it, and at the end of the session we put it back inline, and then we do the same for the other servers. So this allows us to provide, to do full maintenance and support on our server environment with zero, or close to zero user impact.

Q. Related topic there is how do you upgrade your CSMs? Because you kind of put all your eggs in your one basket, right?

A. You still have to upgrade the CSM maybe once a year or twice a year for whatever reason. And again, here we kind of leverage the capabilities. So there is fault tolerance and we have administrative control to basically say the active one. So while the primary one is active we go to the standby one, upgrade the software, make that one active. We have stateful failover anyway so we don't lose any state as we failover. Upgrade the first one again and fail back over. So that's the way we go about upgrade. Thank you Koen. I'm afraid that's all the time that we've got for questions today. And for more information about technologies and solutions deployed at Cisco you can go to the Cisco IT at-work website.

MORE RESOURCES

Where you can find case studies with information about what we deploy, what benefits we've gained, what lessons we've learned, and some operational practices and presentations to help you learn more. Below that you'll see a toll-free number you can call for more information or to place an order. And you can order Cisco resources on the web from the URL at the bottom of this page. I'd like to thank those of you watching for spending this time with us and for being interested in what the Global Technology Seminar Series is all about. We hope that you've enjoyed this seminar and that it's helped answer some of your questions about content switching. And thank you Koen for sharing your time with us and sharing your expertise and enthusiasm for content switching.

CISCO LOGO SLIDE

It was a pleasure Kevin. Thanks for joining, and see you soon.

Corporate Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 526-4100

European Headquarters
Cisco Systems International BV
Haarlerbergpark
Haarlerbergweg 13-19
1101 CH Amsterdam
The Netherlands
www-europe.cisco.com
Tel: 31 0 20 357 1000
Fax: 31 0 20 357 1100

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
www.cisco.com
Tel: 408 526-7660
Fax: 408 527-0883

Asia Pacific Headquarters
Cisco Systems, Inc.
168 Robinson Road
#28-01 Capital Tower
Singapore 068912
www.cisco.com
Tel: +65 6317 7777
Fax: +65 6317 7799

Cisco Systems has more than 200 offices in the following countries and regions. Addresses, phone numbers, and fax numbers are listed on the Cisco.com Website at www.cisco.com/go/offices.

Argentina • Australia • Austria • Belgium • Brazil • Bulgaria • Canada • Chile • China PRC • Colombia • Costa Rica • Croatia • Cyprus • Czech Republic
Denmark • Dubai, UAE • Finland • France • Germany • Greece • Hong Kong SAR • Hungary • India • Indonesia • Ireland • Israel • Italy
Japan • Korea • Luxembourg • Malaysia • Mexico • The Netherlands • New Zealand • Norway • Peru • Philippines • Poland • Portugal
Puerto Rico • Romania • Russia • Saudi Arabia • Scotland • Singapore • Slovakia • Slovenia • South Africa • Spain • Sweden
Switzerland • Taiwan • Thailand • Turkey • Ukraine • United Kingdom • United States • Venezuela • Vietnam • Zimbabwe

Printed in the USA -337355-00 02/06