APPLICATION-AWARE ROUTING IN **SOFTWARE-DEFINED NETWORKS**

SARO VELRAJAN
Director of Technology, Aricent

Aricent

APPLICATION-AWARE ROUTING IN SOFTWARE-DEFINED NETWORKS

Service providers are constantly exploring avenues to both optimize their CAPEX/OPEX and differentiate themselves by improving the user experience. Software-Defined Networking (SDN) presents an opportunity to these service providers by offering a programmable interface to control their networks based on the underlying applications. This paper discusses Application-aware routing in the context of SDN. With the SDN controller applications, it is possible to dynamically provision the network switches based on application characteristics and requirements, leading to a better overall user experience and reduction in bandwidth wastage.

Introduction

Traditional networks were designed to forward packets from source to destination using the shortest route possible. Routers and switches were mostly agnostic to the applications being served by the network, while the application delivery infrastructure was loosely coupled with the packet forwarding infrastructure. In many cases, static network configuration was used to meet the network bandwidth and latency requirements of applications.

Until now, innovations in networking were mostly around increasing the throughput or packet forwarding capacity of network devices such as routers and switches. Then, software-defined networks (SDN) changed the way networks were designed, developed, and deployed, providing significant opportunities for service providers to integrate their application delivery and packet delivery infrastructure, optimize their network architecture, launch newer services, and increase monetization.

Today's networks are forced to be application-aware, to improve user experience, provide service differentiation, and reduce operational costs. Application awareness is the result of gained intelligence about Layer 4 to Layer 7 protocol attributes and delivery requirements. The following are some use cases where the network has to be application-aware:

> A user watching videos requires more network bandwidth compared to a user browsing the Internet. The network has to dynamically adjust the bandwidth allocation to users based on the application, instead of statically allocating capacity.

> Gaming applications require lower network latency compared to applications transferring log files over the network. The network has to adapt to the application's latency needs.

> A router forwarding request to a content server (cache or streaming server) doesn't know whether the content resides in the content server. This increases the number of hops a request has to traverse before reaching the content server that can serve the request.

There are many such use cases that require the networks to be application-aware. This whitepaper describes how SDN enables application-aware routing in service provider networks.

Challenges of Request Routing in Today's Service Provider Network

In today's service provider network, requests are routed to the appropriate content servers using a variety of methods. The application delivery infrastructure is loosely coupled with the packet forwarding infrastructure, resulting in duplication of functions in the network, thereby impacting user experience and increasing operational costs. The following section talks about the various request routing methods in a service provider network.

POLICY-BASED ROUTING (PBR)

Policy- Based Routing (PBR), or filter-based forwarding (FBF), mechanisms are used in the network edges to steer the traffic to the content servers. Service providers define explicit routing policies in routers to forward application requests to a specific content server. For example, routing policies are defined to redirect all HTTP requests (i.e., port 80 traffic) to the Over The Top (OTT) caches sitting in the service provider network that cache Web/video traffic. The limitations of this approach are:

> Routers are unaware of the presence of content in the caches when forwarding the request to the cache. When there are cache misses, latency noticed by applications will be high, especially when handling dynamically generated and non-cacheable content.

> Not a scalable approach when selectively redirecting requests for thousands of Web sites. For each domain to be cached, a separate routing policy must be configured.

BGP ROUTE ADVERTISEMENT

Today, Internet service provider networks have caches from content-delivery network (CDN) service providers such as Akamai, Limelight, and Edgecast. These CDN edge caches advertise routes via Border Gateway Protocol (BGP) for the Web sites that are handled by the CDN. Edge routers learn such routes via BGP and start forwarding requests destined to these sites to the CDN service provider caches. The following are limitations of this approach:

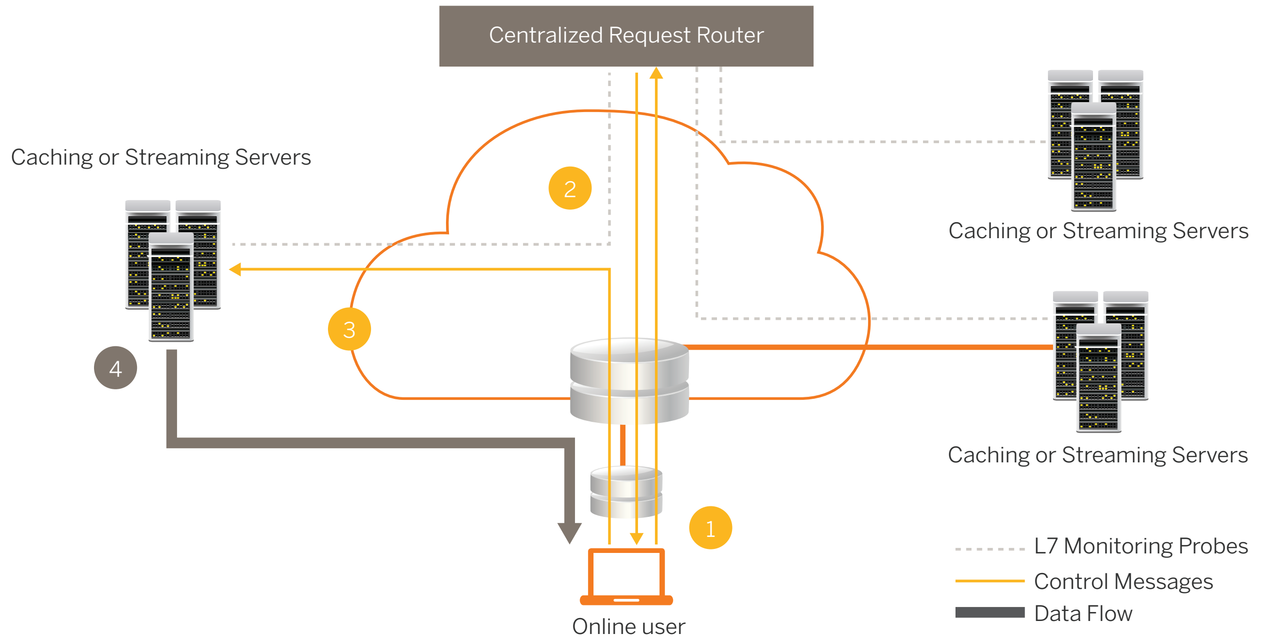
> Traffic belonging to all applications is forwarded to the CDN service provider cache, not just the Web traffic.

> With this approach, service-differentiation-based user requirements cannot be implemented. Edge router do not provide differentiated services for users based on their subscription. For example, a premium subscriber may not get a higher quality of service for live streaming traffic that gets served from the CDN edge cache.

> If an edge router fails, the current flows are not seamlessly routed through it. For example, a user streaming a video will have to buffer the video whenever the edge router fails and does a switchover in the middle of streaming video, which significantly impacts the user experience during the active session at the time of switchover.

CENTRALIZED REQUEST ROUTER

A centralized request router is deployed in the network to redirect user requests to the content server that can best serve the request. The centralized request router is typically an L7 load balancer that redirects requests based on users'



Request Routing Architecture (without SDN) geographical location, availability of content in the content server, service availability, and content server load. The L7-requested routing infrastructure is independently built and maintained to complement the L3 routing infrastructure.

The following are limitations of this approach:

- There are no mechanisms for the centralized request router to communicate with the L3 router to dynamically increase the bandwidth availability for a given user session. The network is always provisioned for peak bandwidth usage, resulting in unused bandwidth.
- Latency requirements of the application are not taken into consideration by L3 routers when routing the requests.
- CAPEX is increased as a result of maintaining a dedicated infrastructure for L7 request routing in addition to the existing routing infrastructure.

Application-Aware Routing Using Software-Defined Network

Software-defined network (SDN) architecture allows service providers to build networks with increased application awareness, which can be built into the network by developing SDN controller applications that keep track of application-level characteristics and use that intelligence to provision flow into the network switches.

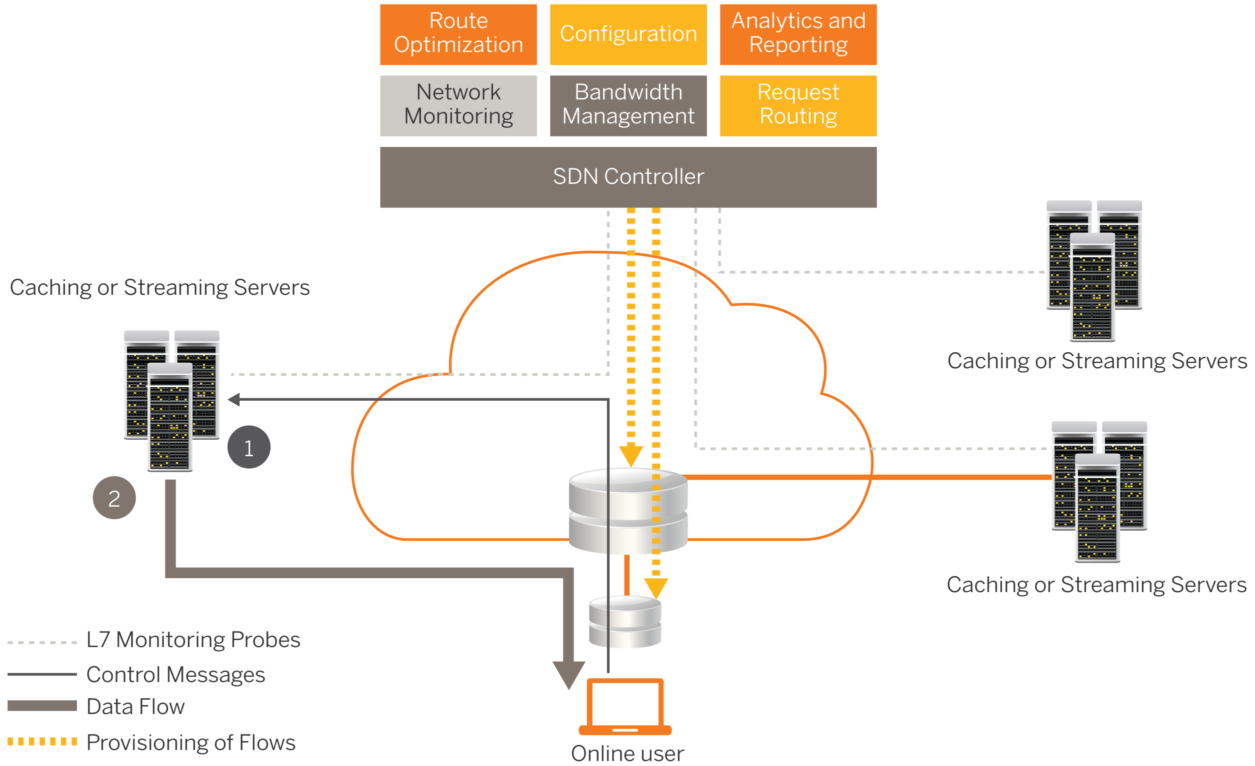
Some of the application-level characteristics and intelligence that can be leveraged for building content-routing applications are:

SERVICE AVAILABILITY

Traditional network monitoring services check only the L2 link or L3 network path availability. However, there may be instances when the content-delivering application could be down or unavailable. For example, an Apache server delivering Web content may not be running, or the content server may not be capable of delivering the content using the requested protocol; or a user may request a video using HTTP protocol despite the server being capable of delivering content only using RTSP protocol. Content-routing applications can be designed to perform service-availability checks before provisioning flows in the network switches.

CONTENT AVAILABILITY

A content-routing application can check the availability of content in a content server (which can also act as a cache) before routing requests to the server. In a CDN, content is pre-populated or stored dynamically in caches/content servers based on the content type, cacheability (static vs. dynamic content), popularity, content size, etc. A content-routing application can communicate with caches or content servers in the network to gain intelligence about content availability.



Request Routing Architecture (with SDN)

This intelligence can then be used to route requests to the appropriate server where the content resides. For example, dynamically generated content is mostly non-cacheable and therefore cannot be served by a caching server. Requests for Web sites that generate dynamic content can be routed directly to the origin server instead of to a caching server, which reduces the latency in a transparent caching environment.

BANDWIDTH MANAGEMENT

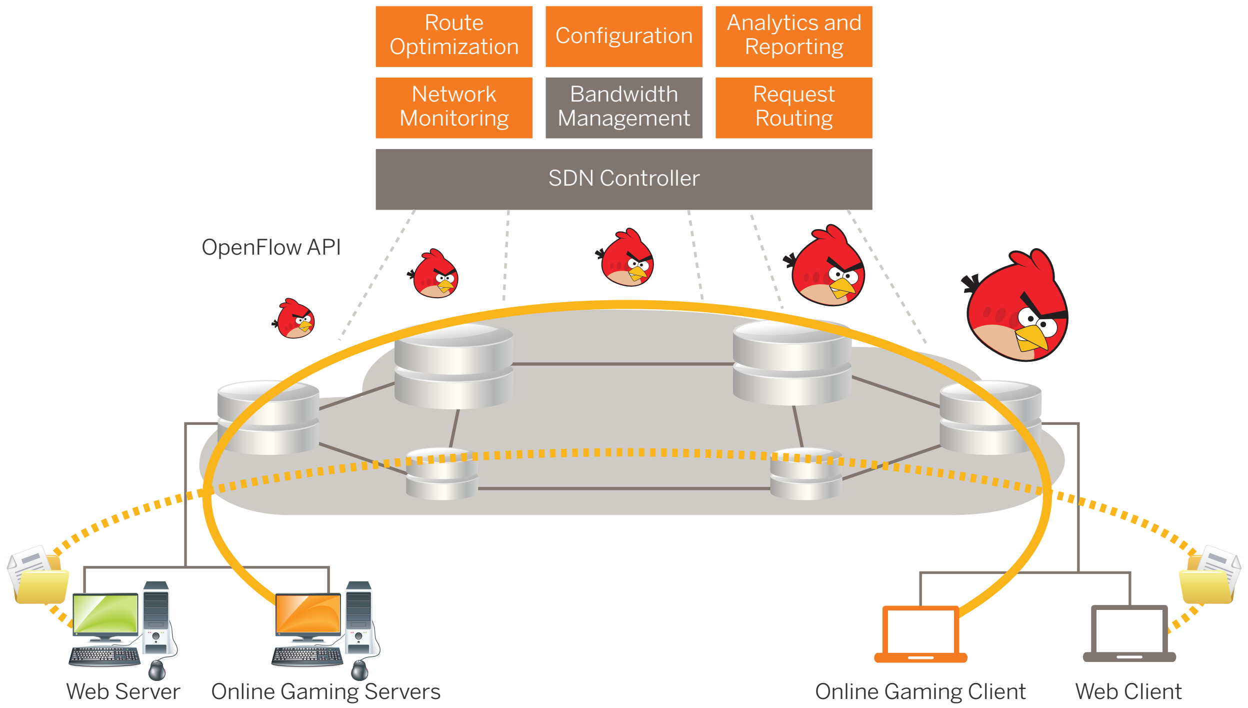
Content-routing applications can periodically monitor bandwidth requirements and re-provision flows in the network switches to match the bandwidth, latency, and QoS requirements of the service. For example, if a premium user is watching online video, the content-routing application can route such user requests through a network path that with a bandwidth capacity and lower latency. This ensures no buffering of video, and therefore a better user experience. When network conditions change (due to congestion, link failure, etc.), the flows can be re-provisioned automatically by the content-routing application so that user traffic is routed through a new optimal network path.

Content-routing applications running on an SDN controller can make routing decisions based on application-level insights and characteristics. This provides a scalable architecture for application-aware request routing. SDN-based architecture eliminates the need for a dedicated or special-purpose content router in the CDN network. It provides granular control for the CDN service providers to influence the quality of experience (QoE) for users and dynamically take corrective actions based on the network conditions.

SDN-based application-aware routing architecture allows service providers to roll out, and monetize from, the newer services. For example, a service provider can provide differentiated services to its customers, like routing a premium user's video streaming session through a network path with higher bandwidth and lower latency. Network conditions are periodically monitored by the SDN controller application to ensure users get assured bandwidth for their streaming sessions, which in turn improves their overall online video watching experience.

SDN-based application-aware routing architecture helps service providers lower CAPEX and OPEX. Typically, service providers size their networks for peak utilization of capacity, resulting in lots of unused bandwidth. In the SDN-based application-aware routing architecture, a live corporate Webcast session, for example, can dynamically request/receive additional bandwidth capacity because service providers are better able to manage their bandwidth capacity, rather than having to allocate a static/fixed bandwidth for each of its enterprise customers.

Application-aware routing using SDN architecture allows service providers to route requests based on content availability, service availability, and application-level bandwidth/latency requirements without losing sight of changing network conditions.



Dynamic Bandwidth Management Using SDN

Conclusion

SDN technology opens up an array of opportunities for network service providers. It helps them roll out monetizable services in their networks faster without having to depend on network equipment manufacturers. Application-aware routing architecture using SDN allows service providers to lower CAPEX/OPEX, to roll out monetizable services, and improve the overall end-userexperience.

Aricent's SDN Expertise

Aricent's SDN expertise covers the entire gamut of SDN technologies, including OpenFlow, SDN applications and Northbound APIs. Our comprehensive controller suite includes support for critical functions like operations, administration, maintenance, lawful intercept, and optimized mobile backhaul. Aricent's SDN applications are built on top of the leading commercial SDN controllers. We are committed to supporting these applications on multiple controllers, thus saving our customers the effort of having to integrate across different platforms.

Aricent's OpenFlow client is a portable implementation of the OpenFlow client functionality and is compliant with version 1.3.0 of the OpenFlow switch specification. Our OpenFlow client can be deployed in both pure and hybrid OpenFlow network environments. Aricent's OpenFlow client extends the power of our award-winning L2/ L3 intelligent switching solution (ISS) framework to provide OpenFlow capabilities. Our client framework is compatible with hybrid and pure OpenFlow switches and can be easily integrated into enterprise, transport, and data center networks.

For information on software-defined networks visit **aricent.com/software/software-defined-networking.html**

SARO VELRAJAN
is a director of Technology at Aricent. He has 15 years of experience working for companies such as Cisco, Ericsson, and Juniper specializing in broadband access, WLAN, content delivery, and software-defined networks.
saravanan.velrajan@aricent.com

INNOVATION SERVICES FOR THE CONNECTED WORLD