Cisco DNA Application Experience

Intent-based networking for applications in the enterprise

Executive overview

This paper introduces and describes how the Cisco® Digital Network Architecture (Cisco DNA) provides an intent-based networking solution for managing application user experience in the enterprise.

Today’s enterprises have thousands of applications on their networks, with hundreds of applications hiding within encrypted flows. Many of these applications directly contribute to business objectives, but many more do not.

As such, few network operators are even aware of how much traffic is traversing their infrastructure by application, let alone how well these applications are treated across their networks. Even fewer operators have aligned their business objectives with end-to-end application policies deployed across their networks. This situation is primarily due to the intrinsic complexity of implementing such policies across various places in the network, because tools, mechanisms, and syntax often vary from platform to platform.

Cisco DNA provides a comprehensive architecture to translate, validate, automate, analyze, and assure intent. It enables operators to effectively and expeditiously monitor, manage, provision, and troubleshoot their applications.

To this end, Cisco DNA infrastructure provides powerful hardware and software capabilities to support granular application recognition—of even encrypted applications—as well as flexible and detailed application treatment capabilities. To harness these powerful infrastructure features and simplify their operation, Cisco DNA Center includes Application Policy, which automates the deployment of intent-based application policies in an end-to-end manner over both brownfield and greenfield networks. Complementing Application Policy, Cisco DNA Center also includes Application Assurance to monitor and report how well applications are being treated across the network, and remediate such treatment if and when necessary.
Introduction

The primary functions of an intent-based networking solution include:

- Translation and validation of intent: An operator expresses business-level intent, and such expressions are translated into validated platform-specific configurations.
- Automation: A controller deploys network-device configurations at scale.
- Analytics: The network operational state is continually monitored with telemetry.
- Assurance: The system validates that the expressed intent is delivered through quantitative metrics or recognizes that the intent is not being met and then guides and automates remediation actions.

The Cisco Digital Network Architecture, illustrated in Figure 1, meets all of these requirements for intent-based application networking in the enterprise.

Cisco DNA software capabilities

Cisco DNA delivers intent-based networking for applications through the following three main components:

- Cisco DNA Application Policy (circle 1 in Figure 1) is an application within Cisco DNA Center that solicits business intent and translates and validates this intent by using Cisco Validated Designs for both greenfield and brownfield platforms in the enterprise. Cisco DNA Application Policy also automates the deployment of these validated configurations to network devices.
- Cisco DNA Programmable Infrastructure (circle 2 in Figure 1) programmable hardware enables powerful infrastructure software solutions to recognize and prioritize application traffic, as well as report on application treatment across the enterprise route, switch, and wireless network, including advanced application-recognition capabilities for hundreds of encrypted applications, without compromising privacy or confidentiality.
• Cisco DNA Application Assurance (circle 3 in Figure 1) is an application within Cisco DNA Center that ingests telemetry data from the network and adjacent data sources and performs contextual correlation and analytics to determine the network state in the context of the expressed intent. This application provides assurance by determining whether or not the intent is being met. If it is met, it supplies quantitative metrics to support validation, and if not, it initiates guided remediation workflows.

This paper introduces these components and shows operators how to use them, and also provides extensive underlying technical detail about what these components do to enable intent-based application experience.

Managing applications in Cisco DNA Center

Four tools within Cisco DNA Center manage applications:

• Application Registry provides and customizes details of individual applications.
• Application Sets provide groupings of applications to simplify policy administration.
• Application Policy specifies and deploys the business intent for application treatment.
• Application Assurance reports on the treatment of applications.

This paper discusses each of these tools. The discussion follows the flow presented in Figure 1, where:

• Business intent is solicited, translated, and validated by Cisco DNA Application Policy and deployed, through automation, to the network infrastructure.
• The network infrastructure recognizes and services applications according to business intent, in addition to reporting on this servicing through advanced instrumentation and telemetry.
• Cisco DNA Application Assurance receives and ingests telemetry and performs contextual correlation and analytics to validate that the intent is delivered (or it triggers a remediation workflow).

Application Registry

The Application Registry contains all the details of the known applications in the enterprise, including more than 1400 default applications, as well as any custom applications, as shown in Figure 2.
You can view applications by their names or group them by standards-based traffic classes. Operators do not need to know these traffic classes, but they are explained for reference as follows:

These traffic classes, based on IETF RFC 4594 “Configuration Guidelines for DiffServ Service Classes”, are summarized as follows:

- **Control**: This traffic maintains infrastructure availability—directly or indirectly—and can be further broken down as:
  - **Network control**: This traffic class is intended for network control-plane traffic, which is required for reliable operation of the enterprise network. Example traffic includes Enhanced IGRP (EIGRP), Open Shortest Path First (OSPF), Border Gateway Protocol (BGP), Hot Standby Router Protocol (HSRP), Internet Key Exchange (IKE), etc.
  - **Signaling**: This traffic class is intended for signaling traffic that supports IP voice and video telephony overlay infrastructures. Example traffic includes Skinny Client Control Protocol (SCCP), Session Initiation Protocol (SIP), H.323, etc.
  - **Operations, administration, and management (OAM)**: This traffic class is intended for network operations, administration, and management of traffic. This class is critical to the ongoing maintenance and support of the network. Example traffic includes Secure Shell (SSH) Protocol, Simple Network Management Protocol (SNMP), syslog, etc.

- **Voice**: This traffic class is intended for audio traffic (only). Example traffic includes G.711 and G.729a codecs, as well as the audio components of multimedia conferencing applications such as Cisco Jabber® messaging; WebEx® meeting applications; and Cisco Webex Teams™, Microsoft Skype for Business, etc.

- **Video**: This traffic can be broken down further, depending on whether the video streams are unidirectional or bidirectional and whether or not the flows adapt to network congestion (that is, are “elastic”, per RFC 4594 terminology):
  - **Broadcast video**: This traffic class is intended for unidirectional, inelastic video streams, such as for broadcast TV, live events, video surveillance flows, and similar video flows. Example traffic includes live Cisco Enterprise TV streams, and Cisco IP Video Surveillance.
  - **Real-time interactive**: This traffic class is intended for inelastic, bidirectional (that is, interactive) video applications. Example traffic includes Cisco TelePresence® meeting applications, immersive gaming applications, etc.
  - **Multimedia conferencing**: This traffic class is intended for elastic, bidirectional multimedia collaboration applications. **Note**: Whenever possible, you should separate audio, signaling, and data media sub-components of this class and assign them to their respective traffic classes. Example applications include Cisco Jabber, WebEx, and Cisco Webex Teams, Microsoft Skype for Business, etc.
  - **Multimedia streaming**: This traffic class is intended for elastic, unidirectional (that is, streaming) video applications. Example applications include Cisco Digital Media System Video-on-Demand (VoD) streams, eLearning videos, YouTube, etc.

- **Data**: Data traffic, from an application treatment perspective, can be broken down into two main subgroups, depending on whether or not the applications are user-interactive:
  - **Transactional data**: This traffic class is intended for user-interactive, data applications, also known as “foreground” applications. Users of these foreground applications are waiting for a response—through the network—before they can proceed with their tasks. Therefore, any delays or losses incurred on these flows will directly affect user productivity. Example applications include data components of multimedia collaboration applications, enterprise-resource-planning applications, customer-relationship-management applications, database applications, etc.
  - **Bulk data**: This traffic class is intended for non-user-interactive data applications, also known as “background” applications. These background applications are typically machine-to-machine flows, so any losses incurred on these flows will not really slow down users (furthermore, any packet losses are automatically corrected by TCP retransmissions). Example applications include email messages, backup operations, FTP/Secure FTP (SFTP) transfers, video and content distribution, etc.
Although traffic classes are a standards-based method of grouping applications together, these methods are not the only ones. Additional grouping flexibility is provided through Application Sets, which are described in the following section.

However, before discussing Application Sets, two additional points relating to the Application Registry bear mentioning. First, you can view details about any given application from within the Application Registry, as shown in Figure 3.

Figure 3. Cisco DNA Center—Application Registry—application details

And second, it is within the Application Registry that you can define custom applications by simply clicking the Add Application button shown in Figure 2. This window introduces a sliding panel to define the new custom application, as illustrated in Figure 4. Within this panel, you provide a name for the application, define how to identify it (for example, by URL or by IP/port-level specifics), and describe how to treat the application, with two choices:

- If you are familiar with the standards-based traffic classes (defined previously), you can simply select the appropriate traffic class to service the application.
- You also can simply select any application from the registry (of 1400+ applications) that the custom app is “Similar To”, and then the custom application will receive the same treatment as the specified (similar) application.

These custom-application definition options are illustrated in Figure 4.
To simplify policy administration, it is helpful to group applications with similar policy requirements together; otherwise you would have to set policy on an application-by-application basis, a process that doesn’t scale well, especially when starting from an application library of more than 1400 applications.

As previously mentioned, grouping by traffic classes is a standards-based method of combing applications with similar policy requirements together. This method is particularly suited to provisioning Quality of Service (QoS) policies to applications. However, although QoS is a foundational and primary use case of application policy, it is not the only type of policy that can be extended to an application. Additional types of application policy include:

- Performance Routing
- Traffic engineering
- Security
- Encryption

Thus, any given method of grouping applications together that serves one of these use cases may not necessarily serve another. For example, although traffic classes may well suit QoS policy, they may not meet the needs of Performance Routing and traffic-engineering policies, etc. Therefore, the manner in which applications are grouped together needs to be flexible and customizable. These parameters embody the purpose of application sets, as shown in Figure 5.
Application sets describe groupings of applications in common business-level language, such as:

- Backup and storage
- Email
- File sharing
- Software-as-service (SaaS) applications
- Software updates
- Others

As such, network operators do not have to familiarize themselves with technical terminology (such as was the case in the previous exercise of defining traffic classes, which included becoming familiar with terms such as “elastic”, “foreground”, “background”, etc.)

Network operators can click the Add Application Set button shown in Figure 5 to create custom application sets, into which they can then drag and drop individual applications to meet their specific application policy requirements.

**Cisco DNA Application policy**

This section discusses Cisco DNA Center Application Policy, which gives you a simple user interface to express your business intent of how your applications should be treated across your networks. It also further details the inner workings of Application Policy, as well as the underlying network infrastructure hardware and software mechanisms that deliver application-policy treatment and reporting.

Only four simple steps are required to deploy an end-to-end application policy across the enterprise network:

1. Click Add Policy and enter a name for the policy.
2. Select a Site Scope to which the policy is to be applied.
3. Express the business intent for the applications.
4. Choose when to deploy the policy (immediately or at a future time).
5. These steps are illustrated in Figures 6 through 10.

**Basic application policy requirements**

The process begins by clicking Add Policy, as shown in Figure 6.

**Figure 6.** Cisco DNA Center—Application policy—Part 1a: Adding a policy

Next, name the policy and select a Site Scope of application, as shown in Figure 7.

**Figure 7.** Cisco DNA Center—Application Policy—Part 1b: Naming the policy
When you click Site Scope, a sliding panel is presented, so you select which site(s) the application policy should be applied to. Sites are presented in a hierarchical manner, and you can select them at any level of the hierarchy. You also can apply the policy globally (by checking the Global box). Specifying a Site Scope is illustrated in Figure 8.

**Figure 8.** Cisco DNA Center—Application policy—Part 2: Selecting the site scope

The third step is to assign business intent to applications. You simply drag and drop application sets to the appropriate business-relevance bucket.

Business relevance has three levels:

- **Business-relevant:** These applications are known to contribute to the business objectives of the organization. Applications assigned as business-relevant receive a preferential treatment across the network infrastructure. Example business-relevant application sets include backup and storage, collaboration apps, database apps, email, network control, network management, signaling, etc.

- **Default:** These applications may or may not contribute to business objectives or there is no business reason to justify explicit policy treatment—whether preferential or deferential—for these applications. As such, these applications remain treated exactly as they were prior to any Application Policy deployment, which is commonly referred to as a "best-effort" treatment and represents a neutral treatment (that is, neither preferential nor deferential). General browsing is an example of a type of application that at times may contribute to business objectives but at other times may not.

- **Business-irrelevant:** These applications are known to have no contribution to business objectives because they are often consumer- and/or entertainment-oriented in nature. Applications assigned as business-irrelevant receive a deferential treatment across the network infrastructure. Example business-irrelevant Application Sets include consumer gaming, consumer media, consumer social networking, etc.

Application Sets are auto-assigned to given business-relevant buckets, but you can modify these sets to suit your business objectives. For example, by default the Software Updates Application Set is assigned to as default business-relevant bucket. However, some companies have found these software updates to be disruptive to their business operations and have chosen instead to assign them to the business-irrelevant bucket, as illustrated in Figure 9.
After business intent is expressed, administrators can optionally choose to Preview policies (as shown at the bottom of Figure 7) before they are deployed to the network devices, or they can proceed directly to Deploy (also shown at the bottom right of Figure 7) these policies.

If Preview is selected, a sidebar will open to present the configurations, on a per-device basis, which the administrator can preview, as shown in Figure 10.

Figure 9. Cisco DNA Center—Application policy—Part 3: Expressing business intent for applications

Figure 10. Cisco DNA Center—Application Policy—Option: Preview and Precheck policies
The administrator can view the configuration that Cisco DNA Center pushes to each device by selecting Generate next to the device (and then View after it is generated, as illustrated in Figure 11.

**Figure 11.** Cisco DNA Center—Application Policy—Option: Preview configurations

You also can elect to Pre-Check the policy by clicking the button. This function performs extensive validation before any configuration is pushed to the device. For example, this function checks and validates:

- Device hardware capabilities
- Device software capabilities
- Feature licensing
- Resource constraints
- Existing policies

Essentially this function looks for anything that may interfere or conflict with the intent expressed in the current policy, and it all happens before any actual deployment to the device. Figure 12 shows the results of a Pre-Check operation; the figure shows that a successful deployment is expected on all the network devices.

© 2018 Cisco and/or its affiliates. All rights reserved.
The final required step is to choose when to deploy the application policy. When you click Deploy, a sliding panel presents the option to Deploy Now or at a future scheduled time, as displayed in Figure 13.
That is all it takes to deploy end-to-end application policies across routers, switches, and wireless platforms in the enterprise network.

Customers have shared with Cisco that deploying such policies has taken them 4 to 6 months and has cost as much as a million dollars per deployment. Then a few months later, as business requirements evolve, the process needed to be repeated. Now, instead of taking months, comprehensive application policies tightly aligned to evolving business objectives can be deployed in a matter of minutes.

Although these four steps are all that is required to deploy policies, Application Policy also supports a few optional steps that you can take to customize application policies to a specific business environment. These optional steps include customizing a:

- Queuing profile
- Marking profile
- Service provider profile

These customizable profile options are discussed in the following sections.

**Queuing profile**

The queuing profile allows an advanced administrator to assign custom bandwidth allocations to traffic classes, as illustrated in Figure 14.
You can increase bandwidth to a traffic class by clicking the + icon for the class; similarly, you can decrease bandwidth from a traffic class by clicking on the – icon for the class. When you add or subtract bandwidth from a traffic class, it has to come from somewhere, so the other classes are subtracted from (or added to). Thus, at some point the administrator can elect to lock the bandwidth allocation to a traffic class by clicking the lock icon; at this point, the administrator can neither add to nor subtract from bandwidth for this traffic class. To comply with best-practice recommendations, some traffic classes are locked by default (such as the network control, signaling, etc.); however, the administrator can override and unlock them according to business requirements.

**Marking profile**

You can modify not only bandwidth, but also the Differentiated Services Code Point (DSCP) markings that Application Policy uses. Although these DSCP markings are standards-based (originating from RFC 4594), some advanced administrators may choose to use different DSCP markings. They can make this decision to accommodate older environments or to support migrations. In either case, the administrator can change DSCP marking profiles from the same sliding panel by choosing Show DSCP Values (instead of Bandwidth Values), as is illustrated in Figure 15.
Service provider profile

As with queuing and marking profiles, you also can customize Service Provider Profiles to modify and optimize enterprise-to-service provider QoS, as illustrated in Figure 16.
These required and optional steps to manage application policies may seem relatively simple and straightforward, and that is intended. However, it may be of interest to advanced administrators to gain a deeper understanding into the innerworkings of Application Policy, as well as the underlying hardware and software infrastructure tools and technologies used to deliver these Application Policy treatments, and these topics are discussed in the following sections.

**Examination of Cisco DNA application policy**

Cisco DNA Application Policy translates business intent into network QoS policy by using a declarative model as opposed to an imperative model.

A **declarative** model focuses on the intent or what is to be accomplished, without describing how it is to be accomplished. For example, a network operator may express that an application such as Cisco WebEx is business-relevant, meaning that it is to be treated with the appropriate service, but the operator does not specify the details of how the QoS/Quality of Experience (QoE) policies should be configured in order to achieve this intent.

In contrast, an **imperative** model focuses on the execution of the intent (describing in detail how the objective is to be realized). For example, an imperative policy may include assigning the Cisco WebEx application to a hardware priority queue with a given bandwidth allocation percentage on a specific network switch interface.

The User-Interface (UI) screens that have been reviewed thus far are the means of soliciting the operator’s declaration of intent. In these screens the operator has specified:

- Which applications contribute to business objectives (that is, are business-relevant)?
- Which applications are known to detract from business objectives (that is, are business irrelevant)?
- Which applications have a neutral impact on business objectives (that is, are default business-relevant)?

Because the UI presents only these three options to the operator (for the sake of maximizing the simplicity of soliciting business intent), some have mistakenly concluded that only three levels of service are provisioned across the network; however, what actually happens in the network is that 12 standards-based classes of service are provisioned across the network, as summarized in Figure 17.

---

**Figure 17. (RFC 4594-based) Application classification, marking queuing, and dropping policies**

<table>
<thead>
<tr>
<th>Business-relevant</th>
<th>Default forwarding</th>
<th>Application examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business-relevant</td>
<td>Default forwarding</td>
<td><strong>Application</strong> class</td>
</tr>
<tr>
<td>VoIP telephony</td>
<td>EF</td>
<td>Priority Queue (PQ)</td>
</tr>
<tr>
<td>Broadcast video</td>
<td>CS5</td>
<td>(Optional) PQ</td>
</tr>
<tr>
<td>Real-time interactive</td>
<td>AF4</td>
<td>BW queue + DSCP WRED</td>
</tr>
<tr>
<td>Multimedia conferencing</td>
<td>AF3</td>
<td>BW queue + DSCP WRED</td>
</tr>
<tr>
<td>Multimedia streaming</td>
<td>AF3</td>
<td>BW queue + DSCP WRED</td>
</tr>
<tr>
<td>Network control</td>
<td>CS6</td>
<td>BW queue</td>
</tr>
<tr>
<td>Signaling</td>
<td>CS3</td>
<td>BW queue</td>
</tr>
<tr>
<td>Ops/Admin/Mgmt (OAM)</td>
<td>CS2</td>
<td>BW queue</td>
</tr>
<tr>
<td>Transactional data</td>
<td>AF2</td>
<td>BW queue + DSCP WRED</td>
</tr>
<tr>
<td>Bulk data</td>
<td>AF1</td>
<td>BW queue + DSCP WRED</td>
</tr>
<tr>
<td>Default forwarding</td>
<td>DF</td>
<td>Default queue + RED</td>
</tr>
<tr>
<td>Scavenger</td>
<td>CS1</td>
<td>Min BW queue (Deferential)</td>
</tr>
</tbody>
</table>

---

© 2018 Cisco and/or its affiliates. All rights reserved.
As illustrated in Figure 17, if you assign an application as:

- Business-relevant: The application is further classified according to the traffic class that the application belongs to (based on the logic discussed earlier in this paper in the “Application Sets” section), including:
  - Voice
  - Broadcast video
  - Real-time interactive
  - Multimedia conferencing
  - Multimedia streaming
  - Network control
  - Signaling
  - OAM
  - Transactional data
  - Bulk data
- Default: The application is marked to Default Forwarding (DF) and treated with best-effort service.
- Business-irrelevant: The application is marked to DSCP Class Selector 1 (CS1), commonly known as Scavenger, and treated with a “less-than-best-effort” service, as specified in RFC 3116.

Cisco DNA programmable infrastructure

At this point in the flow (as illustrated by the first arc in Figure 1), Application Policy has:

- Captured the expressed business intent for the Application Policy
- Translated this business intent into a:
  - Standards-based marking scheme for all the traffic classes of applications (which may be customized, as discussed in the “Marking Profile” section)
  - Standards-based queuing and dropping scheme for all the traffic classes of applications (which, again, may be customized, as discussed in the “Queuing Profile” section)
- Deployed the Cisco Validated Designs configurations to the route, switch, or wireless platforms in the enterprise

The next set of requirements to enforcing application policy across the infrastructure follows:

- Identify the applications on the network.
- Group these applications into the same traffic classes.
- Express the operator-selected business relevance for the applications.
- Mark the traffic end-to-end across the network.
- Ensure consistent congestion management and congestion avoidance of the traffic end-to-end across the network (including internally, where necessary).

To this end, powerful application-recognition software technologies, such as Network-Based Application Recognition 2 (NBAR2), meet these first three requirements. An abstracted cross-platform policy meets the fourth requirement. And hardware platform-specific implementations meet the last of these requirements. Now consider some of these requirements in additional detail to illustrate the powerful role that infrastructure software has in enabling business intent.

NBAR2

Cisco DNA Programmable Infrastructure requires powerful application-recognition capabilities, especially considering that the majority of applications traversing today’s enterprise networks are encrypted. Additionally, application-recognition engines should be extensible (to easily provide support for new applications), consistent (to provide compatible policy enforcement, regardless of platform), and efficient (to avoid placing excessive burdens on the network infrastructure). To meet these requirements, Cisco has developed NBAR2.

NBAR2 is a different technology from the predecessor that shares its name (that is, original NBAR). NBAR2 technology was developed by a startup named P-Cube and so impressed Cisco that the company quickly acquired it in 2004. Other than a shared configuration syntax, NBAR2 has no resemblance to original NBAR technology.
NBAR2 is currently one of the best Deep Packet Inspection (DPI) engines in the industry, providing stateful classification for enterprise business applications, SaaS and cloud applications, as well as hundreds of consumer applications. NBAR2 actually includes thousands of stateful signatures that look into patterns inside packet payload, correlate flows and state, and use sophisticated supplementary mechanisms for identifying applications, including:

- Statistical classification
- Behavioral classification
- Domain Name System (DNS) snooping
- Secure Sockets Layer (SSL) snooping
- Machine learning of applications, services, and servers
- Signature customization

Therefore, NBAR2 can classify more than 90 percent of encrypted traffic on enterprise networks, as can be seen from the following example: In 2017 at Cisco Live Las Vegas, NBAR2 was enabled on the network to classify all the user traffic from the more than 28,000 users in attendance who over the course of 5 days generated more than 65 TB of network traffic, a summary of which is displayed in Figure 18.

**Figure 18.** Encrypted vs. unencrypted breakdown of 65 TB of user traffic from 28,000 attendees at Cisco Live Las Vegas 2017

![Image](image.png)

It should be noted that approximately 72 percent of traffic generated at Cisco Live Las Vegas 2017 was encrypted, as illustrated in Figure 19.
Figure 19. Encrypted vs. unencrypted breakdown of 65 TB of user traffic from 28,000 attendees at Cisco Live Las Vegas 2017

A normal DPI engine would be expected to classify only about 28 percent of the traffic (that is, the unencrypted part). However, NBAR2 was actually able to classify more than 99 percent of all application traffic, recognizing more than 570 applications, as illustrated in Figure 20.

Figure 20. NBAR2 recognized more than 99 percent of application traffic at Cisco Live US 2017 (570 applications)
The total amount of unrecognized applications was less than 1 percent; specifically:

- Percent of HTTPS traffic unrecognized: 0.52
- Percent of SSL traffic unrecognized: 0.34

However, it should be noted that Cisco is developing and adding NBAR2 signatures to the common library every month. Therefore, even these outstanding flows are being analyzed for new application signatures.


Before we look at how NBAR2 accomplishes such detailed and comprehensive application recognition, let’s briefly examine performance impact.

Some administrators have expressed reluctance to enable NBAR2 on their network platforms because of performance-impact concerns (which sometimes have been attributed to experiences with original NBAR technology). Continuing the previous example, when NBAR2 was enabled on a Cisco ASR 1001–HX Aggregation Services Router at Cisco Live US 2017, classifying more than 500 applications at 10-Gbps line rates for 5 days straight, NBAR2 never exceeded 14-percent CPU usage, as shown in Figure 21.

![Figure 21. NBAR2 performance while recognizing 500+ apps at 10-Gbps speeds (less than 14-percent CPU impact)](image)

Now let’s consider how NBAR2 achieves these impressive and efficient application-recognition feats by examining its internal operation.
An important point to understand is that the NBAR2 engines need to process fewer than 5 percent of packets because only the first (or first few) packet(s) of a flow need to be analyzed in order to recognize the application generating the flow. After a classification decision is made, the result is cached in the Flow table (shown at the far left in Figure 22). Thus the remaining (95%+) packets of the flow are already recognized and can bypass all NBAR2 engines (this decision is managed by the NBAR2 Bypass Manager). The combination of the Flow table and the NBAR2 Bypass Manager constitute a “fast path” of application classification, which is highly optimized and can even be implemented in hardware, as is the case for Cisco Catalyst® platforms running the Unified Access Data Plane ASIC, specifically the Cisco Catalyst 3650, 3850, and 9000 Series platforms (discussed in greater detail in the following section).

Also, when an application is recognized, the classification details are cached to the Application table. In this manner, when a new application flow is initiated, full classification processing is unnecessary; rather, you can use recent application-recognition decisions to make a faster and more efficient classification decision for these new application flows.

Of the 5 percent of packets needing additional processing to reach a classification decision, NBAR2 uses three main engines to achieve process them, including the:

- First-in-flow processor: This processor recognizes approximately 40 percent of the packets requiring advanced classification by using a combination of:
  - Custom and/or authoritative (Layer 3 and/or Layer 4) information: You can define custom IP and/or Layer 4 protocol and port combinations to identify your applications. Additionally, you can use technologies such as DNS as an Authoritative Source (DNS-AS), which includes application metadata in DNS records (which are thus considered an authoritative source) for such classification.
- Socket (Layer 3/Layer 4) cache information: By way of review, sockets are combinations of Layer 3 IP addresses, a Layer 4 protocol (TCP/User Datagram Protocol [UDP]) and a Layer 4 port. For example, (10.1.1.1:80 TCP) represents a socket (in this case a web server). When an application server is identified (such as in a previous operation), its details are cached to expedite the classification of new client flows to the same application server.

- Preflow information: For example, a signaling packet can provide information about a Rapid Transport Protocol (RTP) flow that is being initialized. Similarly, FTP control packets can be snooped to identify FTP data flows that are to follow. You can use such details to pre-position NBAR2 to be ready for an immediate classification decision on the subsequent flow, without having to perform additional processing.

- First-payload processor: The first-payload processor recognizes the next 50 percent (approximately) of the packets requiring advanced classification by using a combination of:
  - Custom well-known port payload: When you define a custom protocol, which includes payload details, this custom-definition is checked against the packet.
  - Well-known packet entry: Some applications are bound to “well-known ports” and can be identified from the ports.
  - Heuristic logic: This logic runs heuristics, based on port indicators or regular expressions, seeking to understand which engine in the multipacket processor block should try to handle this traffic and be invoked. So, for example, if the port is 80, it invokes the multiprotocol text parser (which parses HTTP) to process the packet payload.
  - Single packet engine: This advanced engine identifies packets by advanced analysis of the payload.

- Multi-Packet Processor: The remaining 10 percent of packets requiring advanced classification are passed on to the multipacket processor, which uses:
  - A listener that looks for clues in the setup of the session to identify the application
  - Multiprotocol text parser, which examines textual protocols (such as HTTP, SSL, DNS, SIP, etc.) to identify the application
  - Multipacket engine, which examines binary protocols such as RTP to identify the application
  - Statistical machine learning, which provides machine-learning capabilities and behavioral analysis to identify the application
  - Internet Assigned Numbers Authority (IANA) Library, which identifies standards-based protocols

And finally, all these engines use a common Cross-Flow Look-Up table, which shares information from previous decisions to further expedite the application classification process.

When NBAR2 recognizes the applications, they can be checked against the intent-based policies that specify how they are to be treated. This topic is discussed next.

**NBAR2 on Cisco Catalyst UADP Switching Platforms**

One of the incredible advantages of the Unified Access Data Plane (UADP) application-specific integrated circuit (ASIC) is that it is programmable. You can reprogram it with software updates to support new protocols and features in hardware. One such new feature that UADP switching platforms now supports, primarily in hardware, is NBAR2.

NBAR2 became available on UADP Cisco Catalyst switching platforms beginning with Cisco IOS® XE Software Release 16.3. In this implementation, the “fast path” of NBAR2 (called out in Figure 22) is implemented entirely in hardware, providing application recognition for more than 95 percent of packets.

Of the remaining packets, a copy is made of one of the packets, which is then punted to the CPU for deeper analysis (as has been previously described in the “NBAR2 Operations” section). This concept is very important to understand, because at no time is the packet forwarded ever interrupted or delayed on these switches when NBAR2 is used for application recognition.

Before sharing the current performance numbers of NBAR2 operation on Cisco Catalyst 9300 Series
access switches, it may be interesting to review and compare with performance results of the last time Cisco implemented stateful DPI capabilities in hardware. It was done in 2007 on a Cisco Catalyst 6500 daughter card called the Programmable Intelligent Services Accelerator (PISA) module, and it supported 90 protocols at a maximum throughput of 2 Gbps.

By way of comparison, let’s compare NBAR2 performance on a 48-port Cisco Catalyst 9300 Switch:

- Protocols supported: 1400+ (with new applications being added monthly)
- Total number of unique connections per second: 4900, which equates to more than 100 new connections per second per port
- Total number of application flows: 40,000
- Maximum CPU usage: 50 percent

Note: This number is deliberately limited by the architectural implementation.

These numbers indicate that NBAR2 scales very well on Cisco DNA switching platforms.

Additionally, NBAR2 QoS Attributes (discussed in the following section) were added to UADP-based Cisco Catalyst platforms beginning with Cisco IOS XE Software Release 16.8.1. As such, we now can use the same Cisco recommended application classification and marking policy on these switches, with completely identical syntax, as discussed in the following section.

### QoS Attributes

The NBAR2 library has more than 1400 applications; configuring each of these applications with an individual “match protocol” statement would require more than 1600 lines of Command-Line Interface (CLI) (because class maps can match on only 32 match statements, so such a configuration would require hierarchical class maps).

However, as an alternate approach, NBAR2 groups applications by attributes to simplify policy configuration. You can view the attributes of an application in the NBAR2 Protocol Pack documentation or from the command line by using the `show ip nbar protocol-attribute` command. An example of application attributes is illustrated in Example 1, where the NBAR2 attributes of AirBNB are presented.

**Example 1 Displaying NBAR2 protocol attributes**

```plaintext
Router# show ip nbar protocol-attribute airbnb
encrypted     encrypted-no
encrypted     encrypted-no
tunnel        tunnel-no
tunnel        tunnel-no
category      browsing
category      browsing
sub-category  Other
application-group  Other
application-group  Other
p2p-technology  p2p-tech-no
traffic-class  transactional-data
business-relevance   business-irrelevant
```

As of Cisco IOS Software Releases 15.5(3)M and XE 3.16S, every application in the NBAR2 library is assigned two QoS Attributes:

- Traffic class: Based on RFC 4594 application classification logic
- Business relevance: Based on customer usage and operator input

These QoS Attributes for AirBNB are highlighted in Example 1.

The default settings for business relevance are based on customer telemetry data. Specifically, from the telemetry data that customers have elected to share with Cisco (from over a quarter-million network devices), Cisco can identify which applications are commonly considered business-relevant, business-irrelevant, and default. Therefore, the NBAR2 business-relevance attribute has been prepopulated to match. That is why you can see in Example 1 that AirBNB is most commonly identified as a “business-irrelevant” application, because it is consumer-oriented in nature (for the most part).

That being said, the NBAR2 Business-Relevance QoS Attribute is very much intended to be customized by individual network operators to suit their specific business objectives. For example, consider a travel agency that commonly uses AirBNB for its business transactions. So to the travel agency AirBNB would not be considered a business-irrelevant application, but rather a business-relevant one. Such a change can easily be made with an attribute map, as shown in Example 2.
Example 2 Customizing NBAR2 protocol attributes—Changing AirBNB to be business-relevant

Router(config)# ip nbar attribute-map ATTRIBUTE_MAP-RELEVANT attribute business-relevance business-relevant

Router(config)# ip nbar attribute-set airbnb ATTRIBUTE_MAP-RELEVANT

All three options of business-relevance mapping are shown in Examples 3 through 5.

Example 3 Making an application business-relevant

Router(config)# ip nbar attribute-map ATTRIBUTE_MAP-RELEVANT attribute business-relevance business-relevant

Router(config)# ip nbar attribute-set application-name ATTRIBUTE_MAP-RELEVANT

Example 4 Making an application default-business-relevant

Router(config)# ip nbar attribute-map ATTRIBUTE_MAP-DEFAULT attribute business-relevance_default

Router(config)# ip nbar attribute-set application-name ATTRIBUTE_MAP-DEFAULT

Example 5 Making an application business-irrelevant

Router(config)# ip nbar attribute-map ATTRIBUTE_MAP-IRRELEVANT attribute business-relevance business-irrelevant

Router(config)# ip nbar attribute-set application-name ATTRIBUTE_MAP-IRRELEVANT

Any subsequent applications that need to change the default attribute settings for business relevancy would need only a single line to do so (specifically: `ip nbar attribute-set ATTRIBUTE_MAP_NAME`).

“Holy-Grail” classification and marking policy

The addition of traffic-class and business-relevant QoS Attributes to NBAR2 allows for the expression of intent-based classification and marking policies to be reduced from 1600+ lines, to just 57 lines (a 96-percent reduction in configuration complexity). Cisco has dubbed this policy as the “Holy Grail” classification and marking policy, and is shown in Example 6.

Example 6 NBAR2 “Holy-Grail” classification and marking policy

```
class-map match-all VOICE-NBAR2
    match protocol attribute traffic-class voip-telephony
    match protocol attribute business-relevance business-relevant

class-map match-all BROADCAST_VIDEO-NBAR2
    match protocol attribute traffic-class broadcast-video
    match protocol attribute business-relevance business-relevant

class-map match-all REALTIME_INTERACTIVE-NBAR2
```

© 2018 Cisco and/or its affiliates. All rights reserved.
match protocol attribute traffic-class real-time-interactive
match protocol attribute business-relevance business-relevant

class-map match-all MULTIMEDIA_CONFERENCING-NBAR2
match protocol attribute traffic-class multimedia-conferencing
match protocol attribute business-relevance business-relevant

class-map match-all MULTIMEDIA_STREAMING-NBAR2
match protocol attribute traffic-class multimedia-streaming
match protocol attribute business-relevance business-relevant

class-map match-all NETWORK_CONTROL-NBAR2
match protocol attribute traffic-class network-control
match protocol attribute business-relevance business-relevant

class-map match-all SIGNALING-NBAR2
match protocol attribute traffic-class signaling
match protocol attribute business-relevance business-relevant

class-map match-all NETWORK_MANAGEMENT-NBAR2
match protocol attribute traffic-class ops-admin-mgmt
match protocol attribute business-relevance business-relevant

class-map match-all TRANSACTIONAL_DATA-NBAR2
match protocol attribute traffic-class transactional-data
match protocol attribute business-relevance business-relevant

class-map match-all BULK_DATA-NBAR2
match protocol attribute traffic-class bulk-data
match protocol attribute business-relevance business-relevant
class-map match-all SCAVENGER-NBAR2
  match protocol attribute business-relevance business-irrelevant

policy-map MARKING
class VOICE-NBAR2
  set dscp ef
class BROADCAST_VIDEO-NBAR2
  set dscp cs5
class REALTIME_INTERACTIVE-NBAR2
  set dscp cs4
class MULTIMEDIA_CONFERENCING-NBAR2
  set dscp af41
class MULTIMEDIA_STREAMING-NBAR2
  set dscp af31
class NETWORK_CONTROL-NBAR2
  set dscp cs6
class SIGNALING-NBAR2
  set dscp cs3
class NETWORK_MANAGEMENT-NBAR2
  set dscp cs2
class TRANSACTIONAL_DATA-NBAR2
  set dscp af21
class BULK_DATA-NBAR2
  set dscp af11
class SIGNALING-NBAR2
  set dscp cs1
class class-default
  set dscp default

Note: If for some reason an advanced administrator needs to change the marking values for this policy, the administrator can do so, as discussed in the “Marking Profile” section.
The logic of this “Holy Grail” policy follows:

• Every class of traffic uses a “match-all” logical operator (a logical and functional operator), indicating that both match statements need to be true in order for the traffic to match the class.
  - The first match statement matches on the (RFC 4594-based) traffic class.
  - The second match statement matches on whether or not the operator has indicated that the specific application in the traffic class is relevant to the business objectives.

• Any applications indicated to be business-irrelevant are assigned to the Scavenger class.

• Any applications that have not been explicitly assigned a preferential or deferential treatment are treated in the default class.

The benefits of this policy include:

• Because this “Holy Grail” policy is based on abstraction, it significantly simplifies the complexity of configuration, and by extension, monitoring and troubleshooting, of application policies.

• This policy is cross-platform and modular, because QoS Attributes were also introduced on (UADP-based) Cisco Catalyst switches in Cisco IOS XE 16.8.1, and are also planned for Cisco IOS XE wireless LAN controller (WLC) platforms.

• When configured, this policy never has to be changed—even as new applications are added to NBAR2 protocol packs, saving significant operational complexity while simultaneously allowing businesses to be agile in supporting new applications on an ongoing basis.

At this point, the first four challenges of Application Policy have been met (specifically, recognizing, grouping, classifying, and marking applications).

Note: It is beyond the scope of this paper to discuss platform-specific hardware and software QoS architecture; however, for readers interested in Cisco Aggregation Services Routers (ASRs) or Integrated Services Routers (ISRs); Cisco Catalyst switches; or Cisco Wireless LAN Controller or access-point QoS architecture, operation, and configuration, refer to the Cisco Press book “End-to-End QoS Network Design: Quality of Service for Rich-Media and Cloud Networks, 2nd Edition” or the breakout sessions: “Enterprise QoS Design 5.0” and “The Blood and Guts and Gore of QoS,” all of which are referenced in the “References” section of this paper.

Cisco DNA application assurance

Managing network operations manually is becoming increasingly untenable for IT departments. For example, application problems are complex end-to-end problems that can often involve more than a hundred points of failure between the user and the application, as illustrated in Figure 23. Thus troubleshooting these problems can be complicated.
Beyond the complexity induced by the sheer number of variables in the application experience equation, additional network operator challenges include:

- Data-collection challenge: Network operators spend 4x more time collecting data than analyzing and/or troubleshooting based on the insights the collected data reveals.
- Replication challenge: It is impossible for network operators to troubleshoot problems that are not manifest at the same time that they begin troubleshooting (that could be minutes, hours, or days after the reported event); unless operators are able to detect and/or replicate the problem, they are simply unable to investigate it any further.
- Time to resolution: Most network quality problems take hours (or even longer) to find the root case and to ultimately resolve.
- The network is to blame by default: Per customer data the network is often blamed first as the cause of a given problem, but in most instances, this assumption is incorrect, so network operators spend considerable cycles simply proving the innocence of the network.

To address these complex requirements and challenges, Cisco used a top-down design-thinking approach in developing Cisco DNA Assurance. Key features of Cisco DNA Assurance include:

- Be proactive (vs. reactive).
- Provide end-to-end network visibility by continually learning from the network devices and the clients attached to the network.
- Try to get actionable insights to proactively identify and respond to events before users begin complaining.
- Guide remediation actions for more than 150 insights.
- Travel back in time and troubleshoot network problems that have occurred in the past.
- Increase network visibility and enable faster time to resolution by visualizing real-time application traffic flow within in a matter of seconds.
- Develop proactive troubleshooting capabilities to root-cause problems ahead of time and with more granular details.
Cisco DNA Assurance monitors the enterprise using three distinct dimensions of health (as shown in the Cisco DNA Assurance landing page in Figure 24), including:

- Client health
- Network health
- Application health

Figure 24. Cisco DNA Assurance—Landing page

Because the focus of this paper is on applications, let’s look more closely at application health.

**Application health**

Generally when you get on the network you assume it will work properly, until you experience problems with your applications. That is why Cisco DNA Assurance constantly monitors the health of the applications on the network.

The Application health page shows a graphical summary of the health of the applications on the network, as illustrated in Figure 25.

At the bottom of the figure, as with the Network and Client Health pages, is a tabular listing of all applications and their individual health metrics, from which you can select any application by its hyperlink to view the 360 page of the application, as shown in Figures 26 through 28.

**Application 360**

The Cisco DNA Assurance Application 360 page consists of:

- Overall application health score timeline and details, as shown in Figure 26
- Application experience by region, as shown in Figure 27
- Application experience details, as shown in Figure 28
Figure 25. Cisco DNA Assurance—Application health—Application health summary

Figure 26. Cisco DNA Assurance—Application 360—Application health timeline
Figure 27. Cisco DNA Assurance—Application 360—Application Experience (by region)

Figure 28. Cisco DNA Assurance—Application 360—Application experience details
Client 360—Application experience

Each Client 360 view also includes a section that displays the application experience of each application running on that particular client, as shown in Figures 29 and 30.

Figure 29. Cisco DNA Assurance—Client 360—Application experience

Each application that the client is running from any given device is monitored, not only in terms of quantitative metrics (that is, how much application traffic is being generated) but also in terms of qualitative metrics (that is, how well the application is treated over the network). As with other health scores, application health is a composite metric comprising several key performance indicators (KPIs), including latency, jitter, loss, application delay (that is, server delay), etc. These individual KPI details can also be displayed, as shown in Figure 30.
Application experience issues

Cisco DNA Assurance proactively monitors application performance, and if they exceed standards-defined thresholds, it flags poorly performing applications as having issues, as shown in Figure 31.

Figure 31. Cisco DNA Assurance—Application issues
Examining Cisco DNA Application assurance

The question may arise: How are these application metrics measured? And: How is a health score calculated from these measurements?

The answer to the first question is that a variety of tools are available in the Cisco DNA network infrastructure to measure application performance, including:

- Flexible NetFlow (FNF) with Application Visibility and Control: This tool measures application traffic quantity (that is, how many bytes or packets per application flow).
- Perfmon: This tool measures latency and loss of RTP applications such as voice and video.
- Application Response Time (ART) Monitor: This tool measures network and application latency, as well as loss (as reflected by TCP retransmissions) for TCP applications.
- Cisco Class-based Quality-of-Service MIB: This tool measures packets matched, transmitted, dropped, etc. on a per-interface queue basis.
- Cisco IP Service-Level Agreement: This proactive monitoring tool uses probes to test loss, latency, and jitter (among other metrics).
- Application programming interfaces (APIs): These tools connect Cisco DNA Assurance to application servers, clients, and/or agents to share information.

Let’s consider a couple of these tools in more depth. First, consider PerfMon.

PerfMon looks closely at the RTP headers of voice and video packets and makes performance calculations on the metadata contained therein, as shown in Figure 32.

![Figure 32. Calculating jitter and loss from RTP metadata](https://tools.ietf.org/html/rfc3550#section-5.1)

Figure 32 highlights the fact that within every RTP packet there is a sequence number and a timestamp. As packets flow through a network device, gaps in sequence numbers are noted and identified as packet drops. Similarly, when the timestamps of these packets are compared on a packet-by-packet basis (with the sequence numbers providing the packet order), jitter can also be calculated.

These maximum, minimum, and average loss and jitter values are then exported (along with other flow details) in FNF to a NetFlow collector (in this case, Network Data Platform [NDP]).

But what about TCP-based applications? Although it is true that TCP contains a sequence number from which retransmissions can be detected (inferring packet loss), it does not contain any timestamp metadata. So how can latency be measured for TCP?

The answer lies in understanding TCP operations and measuring the gaps between various signals, as illustrated in Figure 33, which illustrates ART operation.
Specifically, ART calculates: [[after second bullet no bullets; should be part of text; the first two tells what ART calculates; then discussion follows]]

- The time delta from the initial TCP SYN to the initial SYN ACK as Server Network Delay (SND), which represents the network delay on the server side of the router.
- The time delta from the initial SYN ACK to the initial ACK as Client Network Delay (CND), which represents the network delay on the client side of the router.

Adding these two delay components together provides total Round Trip Time (RTT): one-way Network Delay (ND) divides this result by 2, specifically ND = (CND + SND) / 2.

Note: Establishing a TCP session with this three-way handshake does not require any application processing; therefore, it provides a baseline of network-only delay.

ART is measured by calculating the time delta between the first response packet and the last request packet.

Removing the Server Network Delay from the response-time result gives an accurate indication of the time it took the application to process the request, which is called the Application Delay (AD).

As with PerfMon, ART includes all these metrics in the FNF export of the application flow to NDP, which performs further analysis and converts these metrics into application health scores.

Application health scores primarily comprise two main components:

- Network QoS: Network QoS is a reflection of how well the application is transported across the network, and it comprises three primary service-level agreements (SLAs), per the industry-accepted standard (IETF RFC 4594):
  - Loss
  - Latency
  - Jitter

---

Figure 33. Calculating network and application delay and loss from TCP flows using ART
• Application Quality of Experience (QoE): Application QoE is a reflection of the end-to-end user experience, which includes not only the network transport, but also the client and/or server experience. As a previously discussed example illustrates, a video application client may adjust resolution and/or frame rates to achieve good results for QoS, but the user experience is downgraded in the process. Therefore, this downgrading of user experience is not captured by QoS statistics; similarly, operations, problems, and errors occurring on application clients and servers may likewise compromise user experience, but are not reflected by QoS statistics.

Therefore, Cisco DNA Assurance monitors both sets of metrics and accurately reflects each in overall application health.

The next step in accurately representing application health is to normalize these metrics into scores that you can compare (and weigh) against each other. For example, if the network reports a value of 213 ms for WebEx latency, is this good? Fair? Bad? What about a measurement of 83 ms for WebEx jitter?

To normalize such KPIs, Cisco has applied standards-based guidance in its application health score calculations. For example, in the case of latency, ITU-T G.114 provides scales that express quality scores for increasing latency thresholds, as shown in Figure 34.

Similar exercises were done for jitter and loss, based on guidance found in ITU-T Y.1541.

Because not every application is equally sensitive to loss and/or latency and/or jitter, an appropriate weighing had to be made on a per-service-class and per-SLA basis. To perform this weighing, again relevant standards were consulted; specifically, the guidance found in RFC 4594 was applied, as summarized in Table 1.
A similar approach was taken for Application QoE metrics, albeit very little standards-based guidance exists in this area. As such, voice, video, and application experts, from both Cisco and customers, were consulted about setting default scoring thresholds for these metrics. With such an approach, application health scores provide a composite metric to network operators about application performance, without overwhelming them with data (and in some cases, noise) that they have to interpret. All underlying metric data is included in 360 views, as has been previously shown.

Summary

This paper details how the Cisco Digital Network Architecture enables intent-based networking for applications in the enterprise.

Specifically, it shows how three key components of Cisco DNA deliver intent-based networking for applications:

- Cisco DNA Application Policy, which solicits business intent and translates the information into Cisco Validated Designs for both greenfield and brownfield route, switch, and wireless platforms in the enterprise; Cisco DNA Application Policy also automates the deployment of these validated configurations to network devices.

Table 1. RFC 4594 loss, delay, and jitter tolerances by traffic class

<table>
<thead>
<tr>
<th>Service class name</th>
<th>Traffic characteristics</th>
<th>Tolerance to</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Loss</td>
</tr>
<tr>
<td>Network control</td>
<td>Variable size packets, mostly inelastic short messages, but traffic can also burst (BGP)</td>
<td>Low</td>
</tr>
<tr>
<td>Telephony</td>
<td>Fixed-size small packets, constant emission rate, inelastic and low-rate flows</td>
<td>Very Low</td>
</tr>
<tr>
<td>Signaling</td>
<td>Variable size packets, some what bursty short-lived flows</td>
<td>Low</td>
</tr>
<tr>
<td>Multimedia conferencing</td>
<td>Variable size packets, constant transmit interval, rate adaptive, reacts to loss</td>
<td>Low - Medium</td>
</tr>
<tr>
<td>Real-time interactive</td>
<td>RTP/UDP streams, inelastic, mostly variable rate</td>
<td>Low</td>
</tr>
<tr>
<td>Multimedia streaming</td>
<td>Variable size packets, elastic with variable rate</td>
<td>Low - Medium</td>
</tr>
<tr>
<td>Broadcast video</td>
<td>Constant and variable rate, inelastic, non-bursty flows</td>
<td>Very Low</td>
</tr>
<tr>
<td>Low-latency data</td>
<td>Variable rate, bursty short-lived elastic flows</td>
<td>Low</td>
</tr>
<tr>
<td>OAM</td>
<td>Variable size packets, elastic and inelastic flows</td>
<td>Low</td>
</tr>
<tr>
<td>High-throughput data</td>
<td>Variable rate, bursty long-lived elastic flows</td>
<td>Low</td>
</tr>
<tr>
<td>Standard</td>
<td>A bit of everything</td>
<td>Not Specified</td>
</tr>
<tr>
<td>Low-priority data</td>
<td>Non-real-time and elastic</td>
<td>High</td>
</tr>
</tbody>
</table>
• Cisco DNA Programmable Infrastructure, which enables the network to recognize and prioritize applications, as well as report on application treatment across the enterprise route, switch, and wireless network.

• Cisco DNA Application Assurance, which ingests telemetry data from the network (and beyond) and performs contextual correlation and analytics to determine the network state in the context of the expressed intent.

This paper introduces these components and shows operators how to use them, and also provides extensive underlying technical detail about what these components do to enable intent-based application experience.

It introduces Cisco DNA Center Application Policy, which provides operators a simple user interface to express their business intent concerning how their applications are to be treated across their networks. Specifically, the basic steps of creating and deploying Application Policy within Cisco DNA Center are detailed, along with UI screenshots showing this workflow. Additionally, optional steps are also presented, including customizing a:

- Queuing profile
- Marking profile
- Service provider profile

Next, this paper details the inner workings of Application Policy, as well as the underlying network infrastructure hardware and software mechanisms that deliver application-policy treatment and reporting. The inner workings of Cisco DNA Application Policy are explained, to show how it used a declarative model (vs. an imperative model) to translate business intent into application classification, marking, queuing, and dropping specifics, all per a standards-based 12-class model.

Next, powerful cross-platform network infrastructure software solutions such as Network Based Application Recognition 2 (NBAR2) show how it can identify more than 90 percent of encrypted traffic in the enterprise.

And finally, the intent-based application experience loop closes with a discussion of Cisco DNA Application Assurance. Cisco DNA Application Assurance ingests network telemetry data and performs contextual correlation and analysis of this data to determine the operational state of clients, network devices, and applications. Cisco DNA Application Assurance monitors multiple application KPIs and interprets them for the network operator, using standards-based guidance, to represent the health of the applications on the network. Furthermore, Cisco DNA Assurance also flags applications that are not meeting recommended performance thresholds as problems, and presents guided-troubleshooting actions to the network operator to remediate them.
References

- A Lower Effort Per-Domain Behavior (PDB) for Differentiated Services: https://tools.ietf.org/html/rfc3662
- Cisco Live Barcelona 2018 BRKCRS-2502: Optimizing Enterprise Networks through SD-AVC: https://www.ciscodive.com/global/on-demand-library/?search=brkcrs-2502#/session/1498024635316001CZfF