Executive overview

The top-of-mind issue in IT organizations today is digital transformation. Recent studies show that 84 percent of organizations are already undergoing a digital transformation journey. This is hardly surprising, as extensive research has shown that digital transformation increases revenues and profitability, and can make or break companies and even entire industries.

The enterprise network lies at the heart of digital transformation. A network that is open, programmable, integrated, and secure maximizes business agility, allowing new business opportunities to be pursued and captured. This is the business value proposition of the Cisco Digital Network Architecture (Cisco DNA), an intent-based networking architecture for the enterprise.

Most networking departments today are bogged down by operations and are spending the majority of their time manually configuring and troubleshooting their networks. In contrast, an intent-based, closed-loop architecture that includes automation and analytics platforms significantly frees up IT time and resources and allows them to be reallocated to driving strategic projects and digital transformation.

This white paper discusses the role of analytics and assurance in the enterprise network architecture and introduces the Cisco DNA analytics platform (an industry first), as well as the powerful network monitoring and troubleshooting application that it enables, Cisco DNA Assurance.

Cisco DNA Assurance monitors the health of clients, network devices, and applications running on the enterprise network. It also expedites the troubleshooting process by leveraging contextual correlation to identify root causes, and then integrates with the automation platform to expedite prescriptive remediation.
Business requirements

According to a 2016 study by McKinsey, companies spend over $60 billion in network operations, with 75 percent of network operating expenses being spent on visibility and troubleshooting.

These statistics are hardly surprising, considering that most enterprises have thousands of users, thousands of applications, and often tens of thousands of network-enabled devices. Furthermore, IP traffic is projected to more than double from 2016 to 2020; additionally, 10 billion more Internet of Things (IoT) devices are expected to come online within the same timeframe (according to Cisco Visual Networking Index™ forecasts).

Managing network operations manually is becoming increasingly untenable for IT departments, a challenge that is exacerbated by the myriad inconsistent and incompatible hardware and software systems and devices in the enterprise. Furthermore, troubleshooting network, client, or application issues is a complex, end-to-end problem that can often involve a hundred points of failure between the user and the application, as illustrated in Figure 1.

Network troubleshooting challenges include:

- **Data collection challenges**: Network operators spend four times longer collecting data than they do on analyzing and troubleshooting using the insights revealed by the collected data.

- **Replication challenges**: It’s impossible for network operators to troubleshoot issues that are not manifesting at the same time that they begin troubleshooting (which may be minutes, hours, or days after the reported event); unless operators are able to detect and/or replicate the issue, they are simply unable to investigate it any further.

- **Time to resolution**: Most network quality issues require hours (or even longer) to identify the root cause and to ultimately be resolved.

- **The network is to blame by default**: The network is often blamed first as the cause of a given problem, but in the majority of instances this is incorrect; as such, network operators spend considerable cycles simply proving the network’s innocence.
Companies that can get a handle on the skyrocketing operational costs of running their networks stand to gain considerable profitability and advantage. The key? Digital analytics. Digital analytics can yield large productivity gains. For example, a 2017 study by McKinsey projected a 20 percent increase in productivity for enterprises using digital analytics to streamline their operations.

This is where the closed loop of Cisco DNA really comes into play, as illustrated in Figure 2.

Figure 2. Cisco DNA: A closed-loop architecture

The primary purpose of the automation platform in Cisco DNA is to “talk” to the network—in other words, to translate the expressed business intent into optimal platform-specific configurations on the network devices. In a complementary manner, the primary role of the analytics platform is to “listen” to the network, specifically to gather, correlate, and make sense of all the network telemetry generated by network devices, in order to correlate this data with the expressed business intent.

Additionally, the analytics platform can provide assurance to the network operator, so as to close the loop with respect to the intent-based networking policies by either:

- Providing quantitative data proving that the intent has been delivered
- Alerting the controller of the need to take a remediation action in order to deliver the intent

This second step can be further complemented by guided and/or automated troubleshooting so as to expedite the remediation process.

Now let’s take a look at the user requirements for analytics and assurance in the enterprise network.
User requirements

Cisco DNA is built on the principles of Design Thinking. Design Thinking is a human-centric approach to innovation that was first taught at Stanford University in the early 1980s as “a method of creative action” and was later adapted for business purposes at the design consultancy IDEO. Design Thinking provides a framework to help teams build products that solve real problems for regular people. Specifically, it is about discovering real human needs and creating solutions that address those needs, with the overall goal being to “surprise and delight” users.

A critical tenet of Design Thinking is that user experience is even more important than technology. This is because technology, solutions, and data tend to change over time; however, intrinsic human needs do not. People, therefore, are the most constant variables in the design equation. As users, we don’t always notice when design is done well, because when form meets function we often simply go about our day. However, we do notice when something is designed poorly, because it confuses us, frustrates us, gets in our way, or slows us down.

As such, Design Thinking focuses on empathizing with users and understanding what they need upfront, so as to build products they’ll love to use—products that should be both useful and usable. At its core, Design Thinking is about solving problems efficiently, effectively, and elegantly. Optimally, Design Thinking lives at the intersection of human needs, business, requirements, and technology solutions, as illustrated in Figure 3.

Figure 3. The Design Thinking “sweet spot”

Therefore, to better understand the requirements of its users, Cisco began its Design Thinking process by conducting dozens of in-depth expert interviews with a wide cross-section of its customers, partners, and in some cases competitors. To maximize neutrality and minimize bias in the interviewing process, the majority of these interviewees were “blind,” in that they were unaware of the party sponsoring the interview (in other words, the interviewees were being asked questions by a neutral third party rather than directly by a Cisco representative).
During this phase, over 60 in-depth interviews were conducted by customers and partners representing some of the largest players in the following industries:

- Banking and financial services
- Manufacturing
- IT companies
- Universities and research centers
- Service providers
- Retailers
- Pharmaceuticals

In addition to these interviews, Cisco regularly meets with its advisory boards of customers, including the Enterprise Technical Advisory Board (ETAB) and Mobility Technical Advisory Board (MTAB). These boards represent Cisco’s largest customers in various industries and provide ongoing feedback and direction for product and technology development so as to meet their evolving business requirements.

Also, teams from Cisco spent time at customer sites performing ongoing observations of how they design, build, and run their networks and the challenges they face in doing so.

Beyond onsite observations, the teams also performed extensive data analysis. For example, while working with Cisco IT, the Cisco DNA team analyzed and correlated over 1455 change management requests and 11,500 incident tickets opened during a 6-month period.

Cisco went to great lengths to better understand who its users were, along with their unmet needs and pain points. Several network personality archetypes began to manifest during this exercise. These include:

- The front-line (Tier 1) network engineer
- The firefighter (Tiers 1 and 2) network engineer
- The expert (Tiers 3 and 4) network architect
- The planner (Tiers 3 and 4) network engineering manager

The point of this exercise was not only to identify the various roles within the networking department, but also to define the responsibilities, needs, motivators, and pain points of each role of network engineer, backed up with direct customer quotes, which are summarized in the following section.

**The front-line engineer**

The front-line engineer is generally the first individual to handle a trouble ticket and is usually the least experienced and least qualified to do so. These engineers have typically received very little training for their role, but at the same time are under a lot of pressure to perform. Obviously, the more issues that can be resolved at this entry-level tier the better, and from 10 percent to 75 percent of issues are reportedly resolved at this stage—illustrating that customers that invest more in training, tools, and resources to support their front line reap corresponding benefits. The Tier 1 front-line engineer user archetype is illustrated in Figure 4.
Figure 4. Tier 1 front-line network engineer archetype

**The front line**

**Ted, 21, Network Ops, T1**

**Role:** 5yrs experience, responsible for general networking support  
**Alliant assurance:** Insurance, 63 locations, 30 VPN locations, 3200 people, 3600 clients  
**Tools used:** Solarwinds, Orion suite, Riverbed, SCC

---

**Time Allocation**

- **Troubleshooting:** High
- **Monitoring:** Basic
- **Planning:** Complex

**Time to resolve:** 15 to 60 minutes

---

**Overview**

- Entry level  
  - Minimum wage  
  - Less skilled  
- Outsourced (sometimes)  
- Remote (often)

---

**Responsibilities**

- Entry-level support  
- IT or network generalist  
- Power user as remote hands (onsite)  
- Commonly outsourced support  
- Rigidly follows a script  
- Opens trouble tickets and routes them to other teams

---

**Needs and motivators**

- Smarter ticketing system through which issues or alerts are routed  
- To interpret and document the reported issue with accurate content and context  
- To validate or re-create the issue in order to better understand it  
- Reliable knowledge base to offer simple fixes  
- Some basic troubleshooting techniques to rule out issues through trial and error  
- A proper escalation path based on severity of business impact, impact of scale, and issue topic  
- Knowledge base to be connected to the alerts received from the tools  
- Ability to resolve the issue call as quickly as possible
Pain points
- Penalized if they don’t put accurate content or enough context into a support ticket
- Penalized if they miss symptoms or never get to the root cause
- Penalized if they don’t use the right tools (which not all of them know how to use)
- Penalized if they don’t escalate to the right expert due to misinformation or limited triage
- Penalized if they don’t escalate in a timely fashion

Quotes
- “Noise gets filtered by the network operations center.”
- NOCs are “typically trained monkeys looking for red lights.”
- Users always complain that it’s a network problem, but “60% of problems end up being application or user issues.”
- Two customers reported that ~10% of issues are resolved at Tier 1; this was the low end of the range.
- Three customers reported that ~75% of issues are resolved at Tier 1; this was the high end of the range.

Summary
The front-line engineer needs some basic troubleshooting techniques to rule out issues through trial and error, because users always complain that it’s a network problem, but “60% of problems end up being application and/or user issues.” Today, front-line engineers aren’t trained on how to use troubleshooting tools, and they are penalized if they don’t use the right tools, miss symptoms, or don’t get to the root cause.

The firefighter
The firefighter represents the first level of escalation. When front-line engineers cannot solve a problem, these are the next set of engineers who see the problem. They are typically better trained and have more experience than their front-line counterparts, but often complain that many of the issues they deal with should have been resolved at the first tier.

Figure 5. The firefighter (Tiers 1 and 2) network engineer archetype
## Overview

- More experience
  - Training
  - Better compensation
- In-house (more often)
- More responsibility, increased access

## Responsibilities

- Primary fixer for simpler Tier 1 and 2 issues
- Basic monitoring and troubleshooting
- Tools
- Outside Network Operations Center (NOC) (mostly)
- Most problems are resolved at this level

## Needs and motivators

- Smarter ticketing system that routes issues or alerts to them with the right context
- Tools that work out of the box without too much customization
- Ability to assess the issue severity, scale, and impact so it can be escalated it to the right expert
- Faster communication and effective collaboration with the right experts for Severity 1 and 2 issues
- Tool to allow them to drill down as deep as needed to resolve issues
- Tool to provide an accurate topology picture to supplement documentation, which could be wrong
- Tool that offers customizable summary views, such as a summary for servers, routers, help desk, etc.
- Real-time status and events

## Pain points

- Severity filter is loose and unreliable
- Fragmented tools and no correlation logic
- Too much raw data, not enough insights
- Dealing with too many issues that should have been fixed at Tier 1

## Quotes

- “I don’t want to be a data scientist; I’m a network analyst.”
- “Documentation isn’t always correct – I need tools to reliably discover and provide me with the correct network view.”
- “Tools must be multivendor; otherwise there are too many to touch.”
- “Tell me if an event is normal.”
- Two customers reported that ~15% of issues are resolved at Tier 2; this was the low end of the range.
- Two customers reported that ~70% of issues are resolved at Tier 2; this was the high end of the range.

## Summary

Firefighters need tools that provide accurate, timely, and detailed information, so that they can drill down as deep as needed to resolve issues, because network documentation is not always accurate and trustworthy. Today, firefighters have to make use of fragmented tools with no correlation logic, which provides them too much raw data and not enough insight.
The expert

The next level of escalation is typically to the expert. These architects have the most training, certifications, and experience but are also trying to focus on forward-thinking planning and thus often resent the constant interruptions that these escalations present. An additional challenge is that much of their expert knowledge is “tribal knowledge” that would be helpful to disseminate to lower support tiers, although this is often difficult or impossible to do, as it’s largely undocumented.

Figure 6. The expert (Tiers 3 and 4) network architect archetype

<table>
<thead>
<tr>
<th>Time allocation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Troubleshooting</td>
</tr>
</tbody>
</table>

**Time to resolve:** 2 to 8 hours

**Overview**

- Most experience
  - Certifications
  - Training/education
- In-house (always)
- Most responsibility: Network design and planning

**Responsibilities**

- High-level alerts and business impacts
- Sophisticated tools and tribal knowledge
- Scenario planning for complex problems
- Big picture and real-time monitoring
- Coordinating large-scale fixes
- Multivendor resolution
- End-to-end visibility
- “The buck stops here”
### Needs and motivators

- Need to first confirm that it’s indeed a network issue, because it’s often misrouted
- Need to simulate the problem with actual data points to assess the issue severity, scale, and impact
- Need to interpret the surface symptoms and issues to get to the root cause
- Need faster communication and effective collaboration with the right experts for Severity 1 and 2 issues
- Need a quick fix or permanent workaround for immediate needs
- Need a good orchestration tool across multiple devices
- Need better capacity planning, visibility, and prediction

### Pain points

- Receives too many issues that should have been fixed at Tier 1
- Constantly disrupted from project work to do troubleshooting
- Spends a lot of time proving it’s not the network

### Quotes

- “It would be great if you could take experience and bundle it into software.”
- “What we’re doing now is unsustainable and is tribal knowledge.”
- “We still need expert eyes to look at tools together with the tribal knowledge to resolve issues.”
- “We need to spend money to understand the trends in order to do things proactively going forward.”
- “At least 50% of time is being spent proving it’s not a network issue.”
- On average, customers reported that ~15% of issues are resolved at Tier 3.

### Summary

Experts need to simulate the problem with actual data points to assess the issue, severity, scale, and impact because they spend a large amount of time proving it’s not a network issue. Today, experts spend more time than necessary troubleshooting issues (which turn out not even to be network issues approximately half of the time); their “tribal knowledge” would help lower tiers of engineers identify the root cause or rule out potential issues earlier in the process.

### The planner

A final role we will consider is the network engineering manager, who is primarily responsible for planning future network deployments. This role spends considerably less time troubleshooting the network and is more interested in baselines, historical trends, analytics, and capacity planning.
Figure 7. The planner (Tiers 3 and 4) network engineering manager archetype

**The planner**

**Sandi, 45, Network engineering manager, T3/T4**

**Role:** 20+yrs experience, responsible for capacity planning for new global branches

**Silicon Valley Bank:** Finance, 48 global locations, 2K employees, 11 ops in T3/T4, 2 DC

**Tools used:** Cisco prime, AlgoSec, SolarWinds, NWatch, mindTree, ExtraHop, DynaTrace

---

### Time allocation

- **Basic**
  - Troubleshooting
  - Monitoring
  - Planning

- **Complex**
  - Analytics
  - Visibility

---

### Responsibilities

- Major capacity planning
- Network design and architecture
- Writing procedures
- Creating trainings for all tiers
- Daily health reports
- Historical trends and analytics
- Tools and process improvements
- Market trend research

### Needs and motivators

- Need better visibility into and analytics on network performance, availability, and users to address issues proactively instead of just reactively
- Need to stay ahead of market-leading products and case studies to evaluate what’s applicable
- Need to upgrade to more robust and correlated tools that enhance the company’s network needs
- Need a tool to audit whether what has been implemented conforms to what was intended
- Need better capacity-planning tools that also do prediction

### Pain points

- Having to wait around for other groups to get their pieces done

### Summary

The planner needs better visibility into and analytics on network performance, in order to solve problems proactively (rather than reactively). Today, planners are forced to wait on other groups to get their respective work done, as they lack the tools and access to gather and analyze the needed information themselves.
Technology requirements

The strategic vision for intent-based networking is to create a self-learning, self-defending, and self-healing network.

While the concept of a self-healing network may seem more like science fiction than reality, if looked at architecturally, it may seem a little closer to reality than perhaps was initially thought.

While Cisco DNA Center is the platform that introduces automation and analytics into the enterprise network, an entire architecture is required to deliver intent-based networking, with the ultimate goal of the self-healing network. This section outlines these technology architectural requirements.

Instrumentation

You cannot monitor, report, analyze, correlate, and learn from something that hasn’t first been measured. As such, instrumentation is the first architectural requirement of a self-remediating network, as shown in Figure 8.

To this end, Cisco has introduced unprecedented instrumentation in its Cisco DNA ready network devices, such as the Cisco Catalyst® 9000 switching family, which is built on the Cisco Unified Access™ Data Plane (UADP) 2.0 Application-Specific Integrated Circuit (ASIC). Each UADP 2.0 ASIC supports 384,000 flexible counters, which could be used to instrument virtually any event within the chip, in addition to the 128,000 NetFlow records that each ASIC can store in its tables.

That being said, it bears pointing out that it’s not only the network that requires instrumentation. For example, to accurately capture user Quality of Experience (QoE), it is critical not only to instrument network Quality-of-Service (QoS) metrics (such as loss, latency, and jitter), but also to include relevant performance metrics from all elements affecting the flow, including hardware and software elements operating beyond the network, such as client endpoints, sensors, application servers, etc.

Clients can also be instrumented. A compelling example of client instrumentation for analytics is Apple and Cisco iOS analytics, in which Apple devices share their view of the network with Cisco DNA Assurance, including the access points they see (complete with signal levels and noise ratios), as well as sharing the reasons with the analytics engine as to why these devices may have disassociated from the network (for example, if the devices have gone idle).

Another valuable example of instrumenting clients is to gain insight into adaptive application behavior. For instance, many multimedia applications use rate-adaptive video codecs. These codecs can raise or lower resolution and/or frame rates, depending on network conditions. Consider the example in which an adaptive video codec...
detects network congestion and—to compensate—lowers video resolution from Full HD (1080p, 1920x1080 pixels) to VGA (640x480 pixels) and video frame rates from 24 frames per second (fps) to 5 fps—the combination of which represents a 97 percent degradation in video quality. After such an adjustment, the network service-level attributes of loss and latency may all be within the highest levels; however, these network metrics alone would not reflect the significantly degraded user experience. As such, a more holistic approach is required to accurately reflect the end-user experience by instrumenting relevant Key Performance Indicators (KPIs) from beyond the network, such as from clients and/or application servers.

Another manner in which beyond-the-network metrics can be instrumented is via sensors. For example, wireless sensors may be deployed throughout the enterprise to proactively and continually test and report on the onboarding experience and the availability and performance of network services (such as authentication, authorization, and accounting [AAA], Dynamic Host Configuration Protocol [DHCP], DNS, etc.), as well as specific applications.

**Distributed on-device analytics**

The wealth of data that can be instrumented on network devices and other devices generates the challenge of deciding which KPIs are more critical than others, so that appropriate actions can be taken in each case. For example, if each of the 384,000 flexible counters on each UADP 2.0 ASIC were equally important and required specific action(s) to be taken, accommodating them could easily flood CPU and/or network resources.

To offset such a scenario, on-device analytics is recommended. In this manner, critical metrics can be identified and immediately acted on, while other noncritical metrics are still retained for informational purposes or for advanced troubleshooting scenarios, which will be discussed later. On-device analytics serves to make the architecture not only more efficient (by distributing analytics and processing), but also more responsive (as the device may self-analyze a problem and immediately remediate it, without having to send telemetry to an external analytics engine and await a response). On-device analytics is illustrated in Figure 9.

**Figure 9. Architectural requirement 2: On-device analytics**

Categorize metrics by degrees of relevance

**Telemetry**

The next challenge is getting data off the device, which requires telemetry. Telemetry has existed in many forms for many years, including legacy protocols such as Simple Network Management Protocol (SNMP), syslog, NetFlow, etc. However, some of these protocols, SNMP in particular, have serious shortcomings in a Cisco digital network architecture. For example, SNMP is poll based, and as such, if a critical KPI was measured on a device, the collector wouldn’t know about it until the next polling interval. Furthermore, the entire MIB would need to be read into the collector, even if only a single data point was required. Finally, if multiple receivers wanted this information, SNMP information would have to be unicast to each receiver. These restrictions make SNMP slow and inefficient for programmable infrastructures.

© 2018 Cisco and/or its affiliates. All rights reserved.
In contrast, model-based streaming telemetry provides significant improvements in getting data off a device. First, data can be “pushed” off a device at any time (rather than “pulled” at given polling intervals). Additionally, individual metrics can be streamed (rather than entire MIBs). For example, if an operator was interested in whether a particular application was being dropped on a router, rather than pulling in the entire CISCO-CBQOS-MIB (which includes all statistics and counters displayed with a “show policy-map interface” command), searching for the relevant queue, and then searching for drops in that queue, the operator can configure the router to push an alert whenever a drop for that specific application is counted. Furthermore, information can be streamed on a message bus, so that any and all parties interested in the data can receive it immediately and efficiently. Figure 10 shows critical metrics being pushed off devices to a collector via streaming telemetry.

Figure 10. Architectural requirement 3: Telemetry

Due to the dynamic nature of digital networks, including their need to make decisions in real time, streaming telemetry is the key to solving these network monitoring and troubleshooting challenges, including reducing (or ultimately eliminating) preventable network outages, as well as detecting urgent situations and automating immediate actions.

**Scalable storage**

The sheer volume of data that a network can generate—in typical cases about 1 TB of data per day—requires adequate provisioning for scalable storage. Scalable storage may take the form of:

- Centralized collectors: Simpler to deploy but the least scalable of the options
- Distributed collectors: Allow for greater scalability but at the cost of complexity
- Cloud-based collectors: Allow for the greatest scalability and are simple to operate; however, they typically entail ongoing expenses from cloud service providers
These storage options are illustrated in Figure 11.

Figure 11. Architectural requirement 4: Scalable storage

Whatever option or options are chosen, network architects will sooner or later have to deal with the question of how much data needs to be stored, and for how long. None of these scalable storage solutions will provide infinite and ongoing capacity (at least, not without significant cost). As such, raw data is typically aggregated after defined time intervals. For example, an operator may decide that after 30 days only averages, along with minimum and maximum outliers, need to be stored (as opposed to all raw data points). The time interval at which granular data is processed to a coarser level will depend on enterprise policy, regulatory requirements, capacity trending and planning requirements, etc.

**Analytics engine**

When the data has been collected from the network into a central or distributed or cloud-based storage system, it is ready for analysis. One of the first analytical operations is to establish a baseline, that is, to trend what is “normal” (or at least normal for a given enterprise network). Once a baseline has been established, anomalies (significant deviations from the baseline) can be identified and trigger further actions, as shown in Figure 12.

Figure 12. Architectural requirement 5: Analytics engine

A key role of the analytics engine is to provide context. In networking, context plays a vital role in understanding what is happening, so as gain accurate insights and to implement correct actions to remediate issues. Accurate context is optimally provided by correlating multiple data sources (akin to multiple points of view) for an object or event in question. Consider the following networking example.

In this example, a user has called the IT help desk and is complaining of poor application performance for Cisco Webex®. The analyst receiving the call has no idea whether this is a network issue or an application issue, nor where the root cause of the issue may be. However, by correlating multiple sources of information, the analyst may be able to zero in on the root cause efficiently and effectively. Figures 13 through 19 show the incremental context and insights that correlating multiple data sources via an analytics engine provide the network analyst who is troubleshooting such an issue.
As a starting point, since the user’s identity is known, the analytics engine can consult the policy engine, specifically the Cisco Identity Services Engine (ISE), for details on the user and the device(s) the user is using, as shown in Figure 13.

**Figure 13.** Contextual correlation example, part 1: Identifying the user and devices

Next, the analytics engine correlates the MAC address of the user’s device(s) with data provided by the DNS, DHCP, and IP addressing system, to identify the IP address of the device experiencing the issue, as shown in Figure 14.

**Figure 14.** Contextual correlation example, part 2: Identifying the client IP address
Following this, the analytics engine correlates the IP address of the device with NetFlow records by filtering records that match the “source address” field of the client device, as shown in Figure 15.

The question now becomes: which flow(s) represent the Webex® application that the user was having issues with? Basic NetFlow records do not include details of which application generated the flow; however, this information can be inserted into NetFlow records via Application Visibility and Control (AVC) technologies. Now the flows can be matched to their respective applications, with the Webex flow being clearly identified, as shown in Figure 16.
Next the analytics engine correlates the network topology that the flow traversed, beginning with the attachment point of the client, as shown in Figure 17.

Figure 17. Contextual correlation example, part 5: Identifying the network attachment point and network topology

Geographical location may also be correlated, as shown in Figure 18, as sometimes issues are local to specific buildings and sites.

Figure 18. Contextual correlation example, part 6: Correlating location information
Following this, device information, including KPIs, policy metrics, configuration details, syslogs, etc., can all be included in the correlation exercise so as to identify the root cause of the network issue, as shown in Figure 19.

Figure 19. Contextual correlation example, part 7: Correlating network device data

In this specific example, a comprehensive contextual correlation exercise accurately identified the root cause of the poorly performing application to be a lack of QoS marking and treatment on the wireless LAN.

However, arriving at a root-cause insight is not enough: the remediation should be actionable. Continuing the example, at this point a remediative action could be presented to the operator, such as enabling the Fast Lane for iOS feature on George Baker’s iPhone and including Webex as one of the whitelisted applications to benefit from this treatment. This recommended remediation action could then be implemented by the analyst via a single click and (in the future, ultimately) automated.

**Machine learning**

As previously noted, analytics engines need to be programmed to compare specific data points with other data points. However, the sheer number of permutations of enterprise network data points is overwhelming for any programmer to identify, let alone program. As such, machine learning algorithms can complement traditional analytics engines in searching for and identifying previously unknown correlations and causations.

Machine learning can be done in a guided or unguided manner (also referred to as supervised and unsupervised). In a guided manner, the machine learning presents observations to the human operator, who then provides input as to whether the observations are interesting or not. If the observation is deemed interesting, then Artificial Intelligence (AI) adjusts its learning pattern to look for similar relationships. In Cisco DNA Assurance, the machine learning algorithm operates in a guided manner and the operator can tag an observation as interesting by clicking a “thumbs-up” icon or, alternatively, deem an observation uninteresting by clicking a “thumbs-down” icon.

© 2018 Cisco and/or its affiliates. All rights reserved.
As previously noted in this paper, in enterprise networking there could be over a hundred points of failure between users and their applications; as such, the permutations of all possible data-point correlations rapidly surpass a human’s ability to identify, understand, and ultimately program them into an analytics engine. For example, with 100 points of failure, the number of potential combinations that could have produced that failure is 4950 \([\binom{n}{2} = \frac{n * (n-1)}{2}]\).

Enter machine learning. Machine learning is an application of artificial intelligence that provides systems the ability to automatically learn from experience and improve without being explicitly programmed to do so. The primary aim is to allow these systems to learn automatically with minimal (or no) human intervention (Figure 20).

The process of machine learning begins with observations of data and with looking for patterns within the data so as to make increasingly better correlations, inferences, and predictions. For example, there may be:

- No pattern between data points
- A pattern between data points that may be attributable to pure coincidence
- A pattern between data points that is indicative of a correlation
- A pattern between data points that is the result of causation

When patterns indicating correlation are identified, AI can search for similar patterns to identify root causes. For example, if a correlation is identified between a particular client operating system and several instances of high wireless roaming rates, AI can search against this pattern for all clients having that operating system, potentially identifying the root cause of poor roaming experience to be in the client OS. However, it is also possible that the correlation may not necessarily be a result of causation, as the root cause may in fact be something else. For instance, the root cause may in fact turn out to be hardware drivers that happened to be released at about the same time as the particular OS version for the client.

Identifying such root causes of issues is the function of cognitive analytics. However, once these root causes have been identified, cognitive analytics can extend into the realm of predictive analytics—in other words, foreseeing imminent issues before they actually occur, with the goal of taking data-driven actions to prevent such issues from ever occurring.

For example, consider a case in which causation has been identified between the memory consumption of a device and its crashing. By identifying this causation, and by carefully monitoring the memory consumption of the family of devices, the system could redirect or prune processes to prevent a system crash before it happens.
Machine learning applied to enterprise networking is illustrated in Figure 21.

**Figure 21.** Machine learning algorithms applied to enterprise networking

Guided troubleshooting and remediation

Analytics and machine learning can help identify an issue or potential issue, but often additional troubleshooting procedures are needed for root-cause analysis.

These procedures typically begin with pulling in additional information from the suspected devices. This is similar to opening a case with the Cisco Technical Assistance Center (TAC), where typically the first step in troubleshooting a reported issue is to supply the TAC engineer with a “show tech” report for the device, which contains virtually every monitored metric.

Following this step, a divide-and-conquer approach is generally required to isolate the problem area, making it smaller and smaller until the root cause is clearly identified. This may include running troubleshooting tests in which the output of one test will determine what the next test should be.

However, zeroing in on a root cause is not the end of the troubleshooting exercise. It’s not enough just to inform the operator of what the cause of the current problem is; rather, specific remediation action(s) should be presented. The operator should be able to quickly enable the recommended action(s) via the automation platform, which pushes out the recommended action(s) to the network device(s). Guided troubleshooting and remediation is illustrated in Figure 22.
Automated troubleshooting and remediation

The guided troubleshooting system needs to monitor its own effectiveness. There may be times when the recommended action does not, in fact, remediate the root cause. In such cases, the guided troubleshooting algorithm too will need to learn and adapt its troubleshooting approach. Perhaps the system may even present its self-measured effectiveness to the operator, stating, for example, that “93% of the time action X has solved this problem.”

Over time, and as the effectiveness score increases, network operators may wish to disengage from their “manual” role in the process (that is, explicitly clicking the button to take the recommended action) and prefer instead for the system to simply “always take the recommended action” when this problem manifests, as shown in Figure 23. At this point, the goal of a self-healing network has been realized.

Figure 23. Architectural requirement 8: Automated troubleshooting and remediation
It’s unlikely that every issue will be addressable by a self-healing workflow, but this is definitely possible for the most common and recurring issues. And the number of addressable issues can be increased over time, via both manual programming and machine learning.

An important observation to make here is that this final architectural requirement involves not only a technical challenge (a self-learning troubleshooting algorithm), but also a confidence challenge.

To illustrate this confidence challenge, consider the self-driving car analogy. Some automobile manufacturers (such as Tesla Motors, for example) produce vehicles that include self-driving capabilities. However, even though these technical capabilities exist, many drivers prefer not to enable them, as they haven’t yet gained sufficient confidence in their abilities.

The situation is similar with networking. Network administrators are responsible for the state of their network. They will allow machines to fix machines only when they (the administrators) have developed adequate confidence and trust in the abilities of their automated systems. As such, it is important to allow maximum transparency of the underlying processes to advanced network administrators so that they don’t view the system as a mysterious black box, but rather as a predictable, deterministic system that they can gradually trust.

Returning to the self-driving car example, if a driver understands that dozens of cameras are continually monitoring every area around the car, in addition to radar, lasers, redundant systems, etc., this will likely go a long way toward helping them gain confidence in their vehicle’s autonomous abilities, and thus increases the likelihood that they will use these features.

Note: The previous sections identified eight distinct architectural requirements to realize the goal of a self-healing, intent-based network. At the time of this writing (April 2018), five of these eight requirements have already been delivered, with a sixth expected within a few months’ time. As such, even though a self-healing network may seem an ambitious goal, it may in fact not be an overly distant one.

Cisco DNA Center

As previously discussed, Cisco DNA Center is a single pane of glass for designing a network, provisioning the network, administering policy for the network, and assuring the network.

To achieve this end, the Cisco DNA Center appliance includes key software components that align to Cisco DNA, including the:

- Network Controller Platform (NCP): The automation component of Cisco DNA
- Network Data Platform (NDP): The analytics component of Cisco DNA

Additionally, like most platforms, Cisco DNA Center includes some built-in applications. One of these applications is Cisco DNA Assurance, which we discuss in detail in this paper. The relationship between NCP, NDP, and Cisco DNA Assurance within Cisco DNA Center is illustrated in Figure 24.
Cisco DNA Assurance

Cisco used a top-down approach to design Cisco DNA Assurance to meet user requirements, including:

- Being proactive, not reactive
- Accurately representing the overall health of all clients, network devices, and applications
- Displaying both wide (end-to-end) visibility and deep (“360-degree”) visibility for any client, network device, or application
- Being network aware with machine learning capabilities
- Using contextual correlation to deliver actionable insights
- Being flexible and programmable
- Providing microservices based on agile streaming telemetry
- Supporting open interfaces and a rich partner ecosystem
- Support a hyper-distributed, multitenant, and cloud-ready architecture
- Providing guided remediation, with the north-star goal of fully automating the remediation process

Key capabilities of Cisco DNA Assurance thus include:

- End-to-end network visibility, achieved by continually learning from the network devices and the clients attached to the network
- Actionable insights to proactively identify and respond to events before users begin complaining
- Guided remediation actions for over 100 insights
- The ability to travel back in time and troubleshoot network issues that have occurred in the past
- Increased network visibility and faster time to resolution by visualizing real-time application traffic flow within a matter of seconds
- Proactive troubleshooting capabilities to identify the root cause of issues, ahead of time and with more granular details
As shown on the landing page (Figure 25), Cisco DNA Assurance monitors the enterprise using three distinct dimensions of health, including:

- Network health
- Client health
- Application health

**Figure 25. Cisco DNA Assurance: Landing page**

### Summary

This paper focused on Cisco DNA Analytics and Assurance. It began by outlining the business and user requirements of analytics and assurance within an intent-based networking architecture.

Following this, it presented an overview of the technology architecture requirements of a self-healing network; these include:

- Instrumentation
- Distributed on-board analytics
- Telemetry
- Scalable storage
- Analytics engine
- Machine learning
- Guided and automated troubleshooting

The majority of these requirements have already been delivered in Cisco DNA, with additional components being added rapidly.

Finally, Cisco DNA Assurance was introduced as an application within Cisco DNA Center to leverage all these technologies to deliver network, client, and application assurance within the enterprise.