**Abstract**

Adaptive automation refers to process in which both the user and the system can initiate changes in the level of automation. The first adaptive automation systems were implemented in associate systems based on models of user’s behaviour and workload. Recently, however, systems have been developed that follow the neural networks approach and use Artificial Intelligence measures to trigger changes in the state of automation. Studies suggested that this approach can facilitate user performance. Further, evidence is beginning to show that people not only think of adaptive systems as co-workers, they may even *expect* them to behave like humans.

Adaptive automation creates new challenges for both users and designers that go beyond traditional ideas of human-computer interaction and system design in run time dynamic network storage cloud.

**Introduction**

The adaptive automation focuses on current research and developments in the neuroscience of information processing and how that knowledge can be used to improve performance in real-world environments with computing systems or information processing with cloud Storage.  Here we need to understand that how the brain processes perceptual and cognitive information can lead to better designs for equipment, systems, and tasks by enabling a tighter match between task demands and the underlying brain processes similar way by using techniques of AI and neural networks Computing systems can process information to store and retrieve data at run time dynamic networked cloud and fetch as and when required.. Ultimately, research in this area can lead to safer and more efficient working solutions.

            Ironically, interest in neural networks and Artificial Intelligence evolved from research surrounding how operators interact with a form of technology designed to make work and our lives easier – automation.  In general, automation is a computing system machine agent capable of doing tasks normally performed by a human being. For example, In a Software Testing environment, either Software testing is done manually or it’s done using some tools and technique using automation designed to reduce task demands and workload. And it allow individuals to increase their scope of operation or control, perform functions those are beyond normal abilities, maintain performance for longer periods of time, and perform fewer mundane activities.

Automation also helps to reduce human error and increase safety.

Research on human interaction with automation concludes that it does not always make the job easier. Instead, it *changes* the nature of work.

“Automation changes the way activities are carried out and can introduce new and different types of problems.  Automation can lead to different types of errors because user system goals may be inconsistent with the goals of systems and subsystems where subcomponents of system are tightly coupled, some weird problems may propagate more quickly and be more difficult to isolate. In addition, highly automated systems leave very less activities for user systems or humans to perform. Consequently, the user becomes a more passive monitor instead of an active participant. It’s shown that this shift from performing tasks to monitoring automated systems can actually inhibit one’s ability to detect critical signals or warning conditions. Further, a user (Technical Person) manual skills can begin to deteriorate in the presence of long periods of automation “

**Adaptive Automation**

            Given the problems associated with automation noted above, researchers and developers have started to find and develop alternative methods for implementing automated systems. Hence idea of *Adaptive automation* developed which has been proposed to address shortcomings of traditional automation. In adaptive automation, the automation level and/or the number of systems operating with automation can be modified dynamically in real (run) time.

“Additionally changes in the state of automation can be initiated by *either* the human (User) or the system (Computing Devices). Consequently, adaptive automation enables the level or modes of automation to be tied more closely to user needs at any given moment”

            Adaptive automation systems can be identified as either *adaptable*or *adaptive. “I we talked about* taxonomy of adaptive technology. One side of this taxonomy concerns the underlying source of flexibility in the system, i.e., whether the information displayed or the functions themselves are flexible. Another side addresses how the changes are invoked.”

In **adaptable** systems, user initiates the changes in the allocation of functions. However, in **adaptive** systems both the user and the system can initiate changes in the state of the system.

            The distinction between adaptable and adaptive terminology can also be described with respect to authority and autonomy.

“There are several levels of automation that range from completely manual, to semiautomatic, to fully automatic.”

With the increase of level of automation, systems adapts more authority and autonomy and at the lower levels of automation, systems by pass autonomy to the user. The user either reject or accept the suggestions and then implement the action.

At moderate levels, the system have the autonomy to process with the suggested actions accepted by the user. And in the end at higher levels, the system may decide on a course of action, implement the decision, and merely inform the user.

Hence higher level of autonomy is expected with system called as run time dynamic adaptive systems.

“With respect to Scerbo’s (2001) taxonomy, adaptable systems are those in which the user maintains authority over invoking changes in the state of the automation (i.e., they reflect a superordinate-subordinate relationship between the operator and the system).” On the other hand in adaptive systems, authority of invocation is shared. Both the user and the computing system can initiate changes in state of the automation.

            There has been discussions having control among modes of operation.  Some argue that users should always have authority over the system because they are ultimately responsible for the behaviour and performance of the system.

In addition, it is possible that users may be more efficient at managing resources when they can control changes in the state of automation.

Many of these arguments are based on work with critical systems related to security of data in which safe and secure operation is of utmost concern.  There may be times when the user is not the best judge of when automation is needed. i.e. changes in automation may be needed at the precise moment the user is too busy to make those changes.

“It can be shown mathematically that the best decisions concerning whether to abort a data transfer are not those where either the human or the Data Center maintain full control? Instead, the best decisions are made when the user and the automation share control.

     Some critical situations where the user system is vulnerable, it would be extremely important for the system to have authority to invoke automation. If situations are at stake or the system is in jeopardy, allowing the system to intervene and circumvent the threat or minimize the potential damage would be paramount. For example, it is not uncommon for many of today's data center/cloud to sustain high bandwidth enough to fulfil high demands. These Conditions make a strong case for system-initiated invocation of automation.

**Adaptive Strategies in run time dynamic systems**

 Let’s discuss about strategies by which adaptive automation can be implemented.

One set of strategies addresses system functionality in which entire process may be allocated to either the system or the user (It may be another system), or a specific task can be divided so that the system and user (client or server) both share responsibility for unique portions of the task.  Alternatively, a task could be transformed to a different format for the user system to perform.

 A second set of strategies is related to the triggering mechanism for shifting among modes of automation. This is a goal-based strategy. In this, levels of automation are triggered by a set of criteria depends on external events/forces.  Thus, the **run time dynamic** system invoke the automatic mode only during specific tasks or in an emergency situation. Another approach would be to use real-time measures of user systems performance to invoke the changes in automation.

A third approach uses models of user system’s (client or server) performance or workload to drive the adaptive logic in run time dynamic systems.

For example, a system could estimate current and future states of a user’s activities, intentions, resources, and performance.  Information about the user, system, and the outside world could then be interpreted with respect to the user’s goals and current actions to determine the need for adaptive aiding. Finally, measures that reflect system workload can also be used to trigger changes among modes.

**Examples of Adaptive Automation Systems**

            Adaptive automation has its beginnings in artificial intelligence.  In the 1970s, efforts were made toward developing adaptive technologies to help allocating and organizing tasks between humans and computing systems.

By the 1980s, researchers began developing adaptive interfaces. For instance, Wilensky, Arens, and Chin (1984) developed the UNIX Consultant (UC) to provide general information about UNIX, information about executing/running UNIX commands, as well as debugging/analysing information. The UC could analyze user queries/questions, confirms the user goals, monitor the user’s interaction history with system, and present the system’s response for better performance. These methodologies are used now a days in business like online shopping to check user interests.

**Workload and Situation Awareness**

***Workload****.* One of the arguments for developing adaptive automation is that this approach can moderate user’s system workload.  Most of the research to date has assessed workload through primary task performance. To perform a simulated task where the object was to send data to targets without colliding with one another.  During manual control, the participants were required to assess the situation on the display, make decisions about which targets to eliminate, and implement those decisions. During a shared condition, the participant and the computer could each perform the situation assessment task. The computer scheduled and implemented the actions, but the operator had the ability to override the computer’s plans. The participants were also asked to perform a secondary task requiring them to monitor the movements of a pointer and correct any deviations outside of an ideal range.  Performance on the secondary task was used to invoke the automation on the primary task. For half of the participants, the computer suggested changes between automatic or manual operation of the primary task and for the remaining participants, those changes were mandated.

            In data storage cloud network shared control resulted in better performance than manual control on the primary task. However, the results showed that mandating the use of automation also bolstered performance during periods of manual operation. Regarding the secondary task, when use of automation was mandated, workload was lower during periods of automation; however, under periods of manual control, workload levels actually increased and were similar to those seen when its use was suggested.  These results show that authority over invoking changes between modes had differential effects on workload during periods of manual and automated operation. Specifically, it’s found that the requirement to “consider” computer suggestions to invoke automation led to higher levels of workload during periods of shared/automated control than when those decisions were dictated by the computer.

***Situation awareness.*** Thus far, there have been few attempts to study the effects of adaptive automation on situation awareness. SA as the ability to perceive elements in the environment, understand their meaning, and to make projections about their status in the near future.  One might assume that efforts to moderate workload through adaptive automation would lead to enhanced SA; however, that relationship has yet to be demonstrated empirically. In fact, within an adaptive paradigm periods of high automation could lead to poor SA and make returning to manual operations more difficult. .

            The effects of a brain-based adaptive automation system on SA. The participants were given a self-assessment measure of complacency toward automation and separated into groups who scored either high or low on the measure. The participants performed a modified version of the MAT battery that included a number of digital and analog displays (e.g., vertical speed indicator, GPS heading, oil pressure, and auto pilot on/off) used to assess SA.

Participants were asked to perform the compensatory tracking task during manual mode and to monitor that display during automatic mode.  Half of the participants in each complacency potential group were assigned to either an adaptive or yoke control condition. In the other condition, each participant was yoked to one of the individuals in the adaptive condition and received the same pattern of task mode switches; however, their own EEG had no effect on system operation. All participants performed three 15-minute trials and at the end of each trial, the computer monitor went blank and the experimenter asked the participants to report the current values for a sample of five displays. Participants’ reports for each display were then compared to the actual values to provide a measure of SA (Endsley, 2000).

Effects of the adaptive and yoke conditions were moderated by complacency potential. Specifically, for individuals in the yoke control conditions, those who were high as compared to low in complacency potential had much lower levels of SA. On the other hand, there was no difference in SA scores for high and low complacency individuals in the adaptive conditions. More important, the SA scores for both high and low complacency individuals were significantly higher than those of the low complacency participants in the yoke control condition. The authors argued that a brain-based adaptive automation system could ameliorate the effects of complacency by increasing available attentional capacity and in turn, improving SA.

**Human-Computer Etiquette**

            Recently, there has been interest in the merits of an etiquette for human-computer interaction. Miller (2002) describes etiquette as a set of prescribed and proscribed behaviours that permit meaning and intent to be ascribed to actions. Etiquette serves to make social interactions more cooperative and polite. Importantly, rules of etiquette allow one of form expectations regarding the behaviours of others. “In fact, Nass, Moon, and Carney (1999) have shown that people adopt many of the same social conventions used in human-human interactions when they interact with computers.” Moreover, they also expect computers to adhere to those same conventions when computers interact with users.

            Miller (2004) argues that when humans interact with systems that incorporate intelligent agents they may *expect* those agents to conform to accepted rules of etiquette. However, the norms may be implicit and contextually dependent: what is acceptable for one application may violate expectations in another. Thus, there may be a need to understand the rules under which computers should behave and be more polite.

            Miller (2004) also claims that users ascribe expectations regarding human etiquette to their interactions with adaptive automation. In their work with the RPA, Miller and Hannen (1999) observed that much of the dialog between team members in a two-seat aircraft was focused on communicating plans and intentions. They reasoned that any automated assistant would need to communicate in a similar manner to be accepted as a “team” player. Consequently, the CIM described earlier was designed to allow users and the system to communicate in a conventionally accepted manner.

             The benefits of adopting a human-computer etiquette are described in a study of human-automation interactions. In particular, they focused on interruptions. In their study, participants were asked to perform the tracking and fuel resource management tasks from the MAT battery. A third task required participants to interact with an automated system that monitored engine parameters, detected potential failures, and offered advice on how to diagnose faults. The automation support was implemented in two ways. Under the “patient” condition, the automated system would withhold advice if the user was in the act of diagnosing the engines or provide a warning, wait five seconds, and then offer advice if it determined the user was not interacting with the engines. By contrast, under the “impatient” condition the automated system offered its advice without warning while the user was performing the diagnosis. Parasurman and Miller referred to the patient and impatient automation as examples of good and poor etiquette, respectively. In addition, they examined two levels of system reliability. Under low and high reliability, the advice was correct 60 and 80 percent of the time, respectively.

            As might be expected, performance was better under high as opposed to low reliability. Further, Parasuraman and Miller (2004) found that when the automated system functioned under the good etiquette condition, operators were better able to diagnose engine faults regardless of reliability level. In addition, overall levels of trust in the automated system were much higher under good etiquette within the same reliability conditions. Thus, “rude” behaviour made the system seem less trustworthy irrespective of reliability level. Several participants commented that they disliked being interrupted. The authors argued that systems designed to conform to rules of etiquette may enhance performance beyond what might be expected from system reliability and may even compensate for lower levels of reliability.

            Parasuraman and Miller’s (2004) findings were obtained with a high criticality simulated system; however, the rules of etiquette (or interruptions) may be equally important for business or home applications.   In a recent study, Bubb-Lewis and Scerbo (2002) examined the effects of different levels of communication on task performance with a simulated adaptive interface. Specifically, participants worked with a computer “partner” to solve problems (e.g., determining the shortest mileage between two cites or estimating gasoline consumption for a trip) using a commercial travel planning software package.  In their study, the computer partner was actually a confederate in another room who followed a strict set of rules regarding how and when to intervene to help complete a task for the participant. In addition, they studied four different modes of communication that differed in the level of restriction ranging from context sensitive natural language to no communication at all.  The results showed that as restrictions on communication increased, participants were less able to complete their tasks, which in turn, caused the computer intervene more often to complete the tasks. This increase in interventions also led the participants to rate their interactions with the computer partner more negatively.  Thus, these findings suggest that even for less critical systems, poor etiquette makes a poor impression. Apparently, no one likes a show-off even if it is the computer.

**Managing Dynamic (Run Time) Adaptive Automation of Storage Cloud.**

            Adaptive automation is also beginning to find its way into commercial and more common technologies. Some examples include adaptive cruise control found on several high-end automobiles and “smart homes” that control electrical and heating systems to conform to user preferences. Also in Managing Data Center and Cloud storage systems and accessing data as and when required.

There are experiences of an adaptive Data Center or Cloud Storage. The Data center is designed to regulate Information processing in a fast and secure way.  The automation monitors the data transfer activities and makes inferences about the data patterns behaviour, predicts future needs, and adjusts the bandwidth or speed accordingly. Here when the automation fails to meet the user’s expectations, the user can set the controls manually.

            The heart of the adaptive Data Storage (or Cloud) is the adaptive control of data center environment and functions to balance two goals: user desires and provide data as and when required instantly without any delay. Because these goals may conflict with one another, the system uses a reinforcement learning algorithm to establish an optimal control policy. The ACCE (Adaptive Control of Cloud Environment) encompasses a learning controller that selects settings based on current states of demand. The controller receives information about an event change that is moderated by a cost evaluator. A state estimator generates high-level information about inhabitant patterns and integrates it with output from an occupancy model as well as information regarding levels of data available to make decisions about changes in the control settings. The state estimator also receives input from an anticipator module that uses neural nets to predict which zones are likely to be inhabited within the next seconds.  Thus, if the data is moving within the center, the ACCE can anticipate the route and adjust the data before it arrives at its destination.            Hence there may be some observations about experiences with adaptive Data center and cloud. First, there will be a hypothetical model of the ACCE’s.  There is a conscious effort to be more consistent in data transfer activities developed a meta-awareness of patterns and recognized behaviour more regular, it facilitated the operation of the ACHE, which in turn, helped it to save energy and maximize service of data access.

            Here the value of communication should be understood. Whenever a bug is noticed in the hardware then system should be modified to broadcast a warning message throughout the Data Center to reset the system. After the hardware problem had been addressed, the warning message should be retained because it provided useful information about what was happened. There may be situations where the user could benefit from being told about consequences of manual overrides.

**Conclusion**

The development of adaptive automation represents a qualitative leap in the evolution of technology. Users of adaptive automation will be faced with systems that differ significantly from the automated technology of today.  These systems will be much more complex from both the users’ and designers’ perspective. Adaptive automation systems will need time to learn about users and users will need time to understand the automation. User and his data center needed some time to adjust to each another.  Further, users may find that adaptive systems are less predictable due to the variability and inconsistencies of their *own*behaviour.  Thus, users are less likely to think of these systems as tools, machines, or even traditional computer programs. Any adaptive Data Storage Center would respond to demand.  **Hence interacting with adaptive systems is more like interacting with a teammate or co-worker**

The challenges facing designers of adaptive systems are significant. Current methods in system analysis, design, and evaluation fall short of what is needed to create systems that have the authority and autonomy to swap tasks and information with their users. These systems require developers to be knowledgeable about task sharing, methods for communicating goals and intentions, and even assessment of operator states of mind.   Researchers and designers of adaptive technology need to understand the organizational, and behavioural patterns that impact communication and teamwork among humans to create more effective adaptive systems. In this regard, ideas regarding human-computer interactions may be a mile stone to the development of successful adaptive systems.

            Thus far, most of the adaptive automation systems that have been developed address life critical activities where the key concerns surround the safety of the operator, the system itself, and recipients of the system’s services. However, the technology has also been applied in other contexts where the consequences of human error are less severe (e.g., Adaptive Data Storage Cloud). Moreover, adaptive automation could be particularly useful when incorporated in systems aimed at training and skill development as well as entertainment.

To date, most of the adaptive automation systems that have been developed were designed to maximize the user-system performance of a single user. Thus, they are user independent (i.e., designed to improve the performance of any operator).  However, overall user-system performance is likely to be improved further if the system is capable of learning and adjusting to the behavioural patterns of its user. Although building systems capable of becoming more user-specific might seem like a logical next step, that approach would introduce a new and significant challenge for designers of adaptive automation – addressing the unique needs of multiple users. The ability of Data Storage Cloud to successfully adapt to demand patterns is due in large part to his being the only inhabitant. One can imagine the challenge faced by an adaptive system trying to accommodate the wishes of two people who want the temperature set at different levels.

The problem of accommodating multiple users is not unique to adaptive automation. In fact, the challenge arises from a fundamental aspect of humanity. People are social creatures and as such, they work in teams, groups, and organizations. Moreover, they can be co-located or distributed around the world and networked together. Developers of collaborative meeting and engineering software realize that one cannot optimize the individual human-computer interface at the expense of interfaces that support team and collaborative activities.  Consequently, even systems designed to work more efficiently based on knowledge of brain functions must ultimately take into consideration groups of people. Thus, the next great challenge for the neuroergonomics approach may lie with an understanding of how brain activity of multiple operators in social situations can improve the organizational work environment.

**Benefits of Run Time Dynamic Adaptive Automation**:

* Cross Browser/OS testing independent of a user's machine
* Testing against older versions of browsers
* Real mobile device testing for mobile web and native apps, both iOS & Android
* Visual Validation for softer "look and feel" testing
* Ability to define test scripts in plain English and translate to any Programing Language code
* Test result reports including screenshots of failed test cases
* Execution from Windows & OSX or Linux desktops
* Execution from CI platforms such as Jenkins & Bamboo

Introduction to Testing:

Cross Device Testing Automation:

There are many automation tools that allows you to run application on multiple platforms and saves a lot of time.

While these automation tools are life-saving but still manual intervention is required because these tools cannot test the usability and accessibility of applications.

Emulator Software: to perform cross platform and environment testing, emulator need to be virtualization. Virtual machine need to be created with different environmental combinations as well as emulator are used to check behavior of applications.

Introduction to Automation Testing:

Introduction to Storage Testing:

Introduction to Networking:

Introduction to Cloud:

Introduction to Cloud Storage:

Adaptive Software Automation with Networked Cloud Computing.

# **What Is the Adaptive Network?**

*The Adaptive Network is a new approach that expands on autonomous networking concepts to transform the static network into a dynamic, programmable environment driven by analytics and intelligence.*

Since the introduction of the first Public Switched Telephone Network, networks have continually evolved. Through the various stages of development—from fixed endpoints in the early Internet to today’s broadband networks that connect mobile users to massive data centers and bandwidth behemoths like Netflix, Amazon, and Facebook—networks have adjusted to accommodate new demands.

The once-static infrastructure is undergoing a more profound transformation than ever before. The latest incarnation is [autonomous networking](https://www.ciena.com/insights/what-is/What-Is-an-Autonomous-Network.html), which is a trend that has been building for some time. The autonomous network runs without much human intervention. It can configure, monitor and maintain itself independently.

But, even though it’s a significant advance, autonomous networking is still too restrictive and too rigid. So Ciena has defined a new approach to the evolution of networking—the [Adaptive Network](https://www.ciena.com/adaptive-network/)—that’s geared toward providing a network that can grow with a company as its business needs and markets change.

### [Introducing the Adaptive Network Vision](https://www.ciena.com/insights/white-papers/Introducing-the-Adaptive-Network-Vision.html)

The Adaptive Network is remaking the network into a dynamic, programmable infrastructure built on analytics and automation.

The Adaptive Network allows providers to evolve their current infrastructures into more of a communications loop that relays information from network elements, instrumentation, users, and applications to a software layer for review, analysis, and action—rather than bogging down the network itself.

The Adaptive Network includes three important layers:

* [Programmable infrastructure](https://www.ciena.com/adaptive-network/#programmable-infrastructure)**:** This includes the network’s physical and virtual elements, as well as the telemetry gathered from them. The programmable infrastructure layer is highly intelligent and interprets data so the network can make decisions—whether that means routing traffic around a circuit that's down or investigating and correcting an issue with latency or lower-than-expected capacity on a specific link. Programmable infrastructure requires a flexible grid; a reconfigurable photonic layer to give the ability to reroute channels of variable spectral occupancy across any path, and across any optical spectrum in the network; and telemetry from the IP layer correlated with routing data. In addition, a programmable infrastructure needs tunable coherent transponders to efficiently map a flexible number of client signals to the variable line capacity. In turn, that requires a centralized purpose-built Optical Transport Network (OTN) or packet switching architecture.
* [Analytics and intelligence](https://www.ciena.com/adaptive-network/#analytics-and-intelligence)**:**The programmable infrastructure produces significant amounts of data. Some of it is big data that indicate trends that the network learns and adjusts for over time. Big data can inform the network on how to adjust in the long term, which traffic patterns to look out for, and which parts of the network could be vulnerable. Then there’s small data—things that are happening at a fairly rapid pace. It could be a flicker on a circuit or an immediate request from a customer. Such events require a speedy response from the network, and those moves will be made by the analytics. But once the decisions have been made, a human operator or pre-defined policies could step in and approve or change things as necessary. In a truly autonomous network, there would be no operator influence at this point.
* [Software control and automation](https://www.ciena.com/adaptive-network/#tabbed-component-headline)**:** Research shows the undisputed number one cause of network outages is human error, with estimates as high as 32 percent, according to Dimension Data's 2014 Network Barometer report. Effective automation of network tasks, such as loading access controllers and provisioning routers, or automated calculation and configuration of TE tunnels to optimize traffic and relieve congestion,can eliminate those errors and keep the network running at peak performance. The ability for automation to work across multiple vendors is critical. Some technologies are good at working with one set of devices from a single vendor, but few networks are built on a single vendor’s gear. Networks have to interoperate, using APIs, to function efficiently and move data efficiently and swiftly from point to point.

The development of the Adaptive Network is a watershed moment for the networking world. It’s a cohesive evolution that supports all aspects of intelligent automation—such as intent-based orchestration, analytics, and programmable domain control. It’s a micro services-based architecture that delivers extensibility and scale. Plus, it takes a DevOps integration approach to provide operational and service agility.

The Adaptive Network is a new approach that expands on autonomous networking concepts to transform the static network into a dynamic, programmable environment driven by analytics and intelligence.

# **4 ways an Adaptive Network can overcome today’s challenges and take your network to the next level**

*In the*[*first blog in this series*](https://www.ciena.com/insights/articles/the-adaptive-network-why-automation-alone-is-not-enough.html)*, we discussed the key aspects of the Adaptive Network, and why, as traffic increases and becomes less predictable, effective partnerships will become more critical. We all know that network management conditions are tough for carriers, with legacy network limitations; and intense competition from new market entrants and evolving business models. An agile and adaptive network can help operators overcome these challenges; and exploit the emerging opportunities like IoT and 5G use cases, says Françoise Pouliquen, Carrier Account Director, Ciena EMEA.*

There is a relentless push-pull from rapid business and technology change affecting operators today. On one hand, dramatic growth in subscriber demands are driving fronthaul and backhaul traffic and putting networks under intense pressure. While on the other, there’s an industry wide race to develop and commercialize new revenue-generating services, such as IoT use cases and 5G mobile services – and to implement the network technologies and architectures needed to support and deliver them. On top of that, new market entrants, including some of the largest internet companies, are deploying massive-scale network connections that support low-cost data transport between key locations and data centers with unrivalled economies of scale.

The challenges for operators are; how to take exponential traffic growth in stride; how to prepare the network for the next-generation of IoT and 5G use cases; and how to remain competitive on price with large connectivity providers in the market.

### [Light Reading webinar: Intelligent Multi-Layer Automation – Mapping the Way to a Self-Healing, Self-Optimizing Network](https://event.on24.com/wcc/r/1813363/E3200CB52652CE2F6BACCE351766A7CE?partnerref=ANLR5social18)

Here are four key ways an Adaptive Network can help:

**1) Increasing network agility and efficiency** – because the Adaptive Network turns the network into a dynamic, programmable infrastructure built on analytics and automation, it helps meet growing bandwidth needs with on-demand scalability. As well as helping operators to handle incremental traffic growth and unpredictable demand peaks, an Adaptive Network supports real-time scaling and resource-allocation to support differentiated QoS for different applications and use cases, paving the way for commercial 5G services.

The agility of an Adaptive Network also helps operators to maximize efficiency by automating a wide range of manual networking processes, from routine service provisioning and turn-up, to resource discovery and traffic routing over the best available components and paths. In this way, it helps operators compete effectively with even the largest connectivity providers.

**2) Future-proofing the network with industry leading packet-optical solutions -**Ciena has created the industry’s most scalable portfolio of programmable, packet-optical network infrastructure to help operators meet massively growing bandwidth demand up to the edge.  Our packet-optical solutions cover the metro edge, between data centres, the backbone core, and submarine. This packet-optical market leadership, enabled by consistently high R&D investment, is based on our deployed 100G, 200G, and 400G capable in-house modem technology, supported by a unique combination of software intelligence to bring the best of optical innovation to the market.

Our leading technology portfolio also supports seamless convergence of voice and data traffic for mobile operators. This will allow operators to integrate 4G and 5G traffic in the future, and to deliver it extremely cost effectively across a unified infrastructure.

By ensuring that the network can keep pace with exponential increases in bandwidth demands and new services requirements (low latency, high availability) our packet-optical portfolio protects our clients’ business for the long-term.

**3) Helping avoid vendor lock-in with open networking -**Many network providers design their portfolios to work together, however, infrastructure is becoming increasingly complex, needing to integrate components and processes in multi-vendor environments. This approach requires large-scale “lift-and-shift” infrastructure upgrades which are costly and disruptive, as well as reducing ROI on existing network equipment.

To maximize cost efficiency and value for our clients, Ciena’s portfolio of hardware and software are designed on the principle of [openness.](https://www.ciena.com/products/) This allows operators to tie an entire network infrastructure together into a single environment that delivers value for the business and end-customers long-term.

As well as integrating all equipment – both legacy and new – Ciena can help monitor and manage multi-vendor networks with a centralized, integrated set of tools. This capability is delivered by [Ciena’s Blue Planet](https://www.ciena.com/blueplanet/products/) software suite and our [Manage, Control and Plan](https://www.ciena.com/products/manage-control-plan/) (MCP) software, which gives full visibility of resources and services across multi-vendor domains, with tools to troubleshoot and manage diverse infrastructure components remotely.

**4) Driving network innovation in strategic partnership** - Finally, but equally importantly, Ciena is a strategic partner for global operators. Based on our financial and operational stability, we are able to commit to continual innovation of our portfolio, ensuring that our clients can embrace emerging opportunities and take future network challenges in our stride.

One example of how we are investing for the future is our recent acquisition of [Packet Design](https://www.ciena.com/insights/articles/Ciena-Completes-Acquisition-of-Packet-Design.html), with network performance management software that is focused on Layer 3 network optimization, topology and route analytics. By integrating Packet Design into Blue Planet, we will be able to extend our intelligent orchestration and automation capabilities from layer 0, 1 and 2 into the IP layer. As a result, our clients will be able to further optimize service delivery and maximize resource utilization – taking the Adaptive Network to the next level.

**Adaptive Interface to Scalable Cloud Storage:**

Many of today’s applications are delivered as scalable, multi-tier services deployed in large data centres. These services frequently leverage shared, scale-out, key-value storage layers that can deliver low latency under light workloads, but may exhibit significant queuing delay and even dropped requests under high load.

Stout is a system that helps these applications adapt to variation in storage-layer performance by treating scalable key-value storage as a shared resource requiring congestion control. Under light workloads, applications using Stout send requests to the store immediately, minimizing delay. Under heavy workloads, Stout automatically batches the application’s requests together before sending them to the store, resulting in higher throughput and preventing queuing delay. We show experimentally that Stout’s adaptation algorithm converges to an appropriate batch size for workloads that require the batch size to vary by over two orders of magnitude. Compared to a non-adaptive strategy optimized for throughput, Stout delivers over 34× lower latency under light workloads; compared to a non-adaptive strategy optimized for latency, Stout can scale to over 3× as many requests.