Capstone Project Report:
A Study of Intent-based Networking

Lihe Fan

University of Alberta

Master of Science in Internetworking

Under the supervision of
Mr. Shahnawaz Mir

Introduction

Intent-Based Networking (IBN) is an automated process supported by some software. Unlike the old manual configuring processes, Deeper intelligence and analysis are involved in Intent-Based Networking. An outcome, so-called intent, can be made by the network administrator, while the administrator does not need to know the exact process of achieving the outcome. Instead, the network software can help determine how to get the outcome using intelligent technology. Compared to the regular network, there are some advantages that an intent-based network provides. IBN reduces the amount of manual operation, improves the ability of analysis, accelerates the speed of troubleshooting, and has better security. In general, it could provide significant timesaving, especially regarding network configuration and reaction to network issues. Intent-based networking depends on artificial intelligence and machine learning to prescribe and perform activities and react to network concerns. The software analyzes the given intent and then finishes the jobs if the intent requires them. With AI technology, intent-based networking has more capability to automate networking tasks than regular software-defined networking. (“What is intent-based networking,” n.d.)

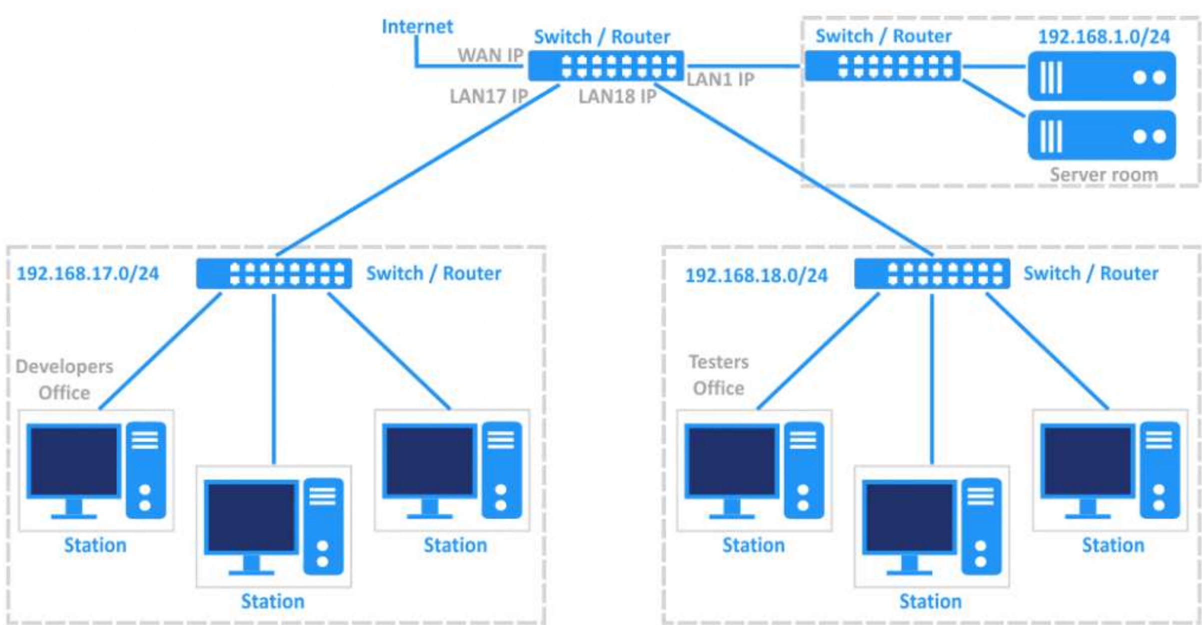
This report aims to analyze and study the design, application, and other aspects of Intent-Based Networking. It compares it with the manual process of network configuring and explores the relationship between IBN and Software-Defined Networks (SDN), then learn about how intent-based networking works with business intent.

Background: Traditional networking and Software-Defined Networking

Traditional networking is the most prevalent form of networking. It controls network traffic via fixed-function, dedicated hardware, and network devices, such as switches and routers. These devices have distinct functions that complement one another and maintain the network. The functions of traditional networking are implemented by dedicated devices, using switches, routers, and application delivery controls, and ASIC and other dedicated hardware are primarily responsible for implementing its functionalities. Scalability is a significant challenge with traditional networks. The majority of switching hardware and software is proprietary, and it is uncommon for providing APIs to be released. For example, if a new device from a different vendor is added to the topology shown in Figure 1, many manual operations on most devices will be involved, which is inefficient. Typically, traditional networks and proprietary provisioning software function well together. Unfortunately, traditional networks are unable to modify this software as needed, and hardware-centric networking can be somewhat restrictive in terms of what it can employ. (“SDN Vs Traditional Networking”, n.d.)

Figure 1

Example of Network Topology



Note. Adapted from *Official NAKIVO Blog* (Bose, M., 2022), website https://www.nakivo.com/blog/types-of-network-topology-explained/

Software-defined networking is defined as the decoupling of the network's control and packet forwarding planes. Software-defined networking enables networks to connect through APIs. This relationship between software-defined networking and APIs promotes application efficiency and safety and contributes to the development of a scalable, flexible network architecture. (“SDN Vs Traditional Networking”, n.d.)

Globally, software-defined networking is extensively utilized by corporations for application distribution since it enables these organizations to deploy their applications rapidly and reduce deployment and running expenses. Software-defined networking enables centralized provisioning and management of network services by IT administrators.

Software-defined networking leverages open APIs to uphold network control, hence yielding network resource efficiency, programmable administration, and management. By decoupling network configuration from their underlying networking equipment, SDN creates network control. This decoupling enables OpenFlow and more open protocols, allowing access to network switches and routers, while most of these devices typically have proprietary and closed firmware. Using the overall software control at the network's endpoint, these open protocols can frequently employ these types of firmware. (“SDN Vs Traditional Networking”, n.d.)

The major difference between traditional networks and SDN is that traditional networks are hardware-based, and SDN is software-based. So that the scalability and flexibility of SDN are enhanced, and SDN can provide greater control and simpler resource management to users, enabling users to use the control plane to manage resources virtually.

In contrast to traditional networks, SDN controllers provide a northbound interface that connects with APIs, enabling software developers to program the network so that they do not have to use traditional networking protocols to program the network. SDN provides a better ability to communicate with network-using hardware. SDN enables users to deploy new devices with software and IT managers to direct network paths and network services.

The virtualization of networks is another feature of SDN. When a network is virtualized via SDN, an abstract copy is created. This virtualization enables resource provisioning from a centralized point. Instead of the physical location of the control plane in traditional networks, virtualization moves SDN's control plane from physical hardware to software-based applications. Virtualization enables the control plane to be accessible via a connected device and provides IT managers with greater control over directing traffic flow via a centralized user interface (UI). This consolidated user interface provides greater control over the configuration and functionality of their networks. The centralized Interface enables users to swiftly manage various network configurations, which is particularly advantageous for network segmentation.

SDN enables IT managers to provision resources and bandwidths, allowing them to grow them as needed without necessitating the investment of additional physical equipment. Traditional networking requires new hardware to expand network capacity, but SDN just requires software operations. So that SDN has become a popular choice to replace traditional networking.

Intent-based Networking Dissection

What is IBN?

Intent-based networking is a software-enabled automation technique that involves artificial intelligence and high-level analytics and orchestration to promote network efficiency. The network system is capable of achieving the desired business outcomes specified by network users by translating users' intents into appropriate configurations. The whole process does not involve any manual operations.

Consider, for instance, the requirement for secure connections between two networks. A broad aim would declare that a secure tunnel is required between Networks A and B. An administrator would specify which types of traffic should use the tunnel as well as any other attributes wanted for the tunnel. However, the administrator does not need to indicate how the tunnel should be established, such as the number of involved devices, how BGP advertisements should be delivered, or which features and parameters should be configured. Based on the service description, an intent-based networking system automatically generates a precise configuration of all devices. Using closed-loop validation, it then constantly verifies the configuration's accuracy and reliability. ("What is intent-based networking," n.d.)

What Can IBN Solve?

Networking has traditionally been driven by manual, CLI-based operations, basic element management systems (EMS), and automation scripts. The majority of network breakdowns are caused by human mistakes during network operations, which has a negative impact on the efficiency of network configuration. In multiple ways, intent-based networking reduces errors and risks while increasing operational efficiencies. Intent objects are abstract representations of the desired network features or results. Before applying intent objects to the network, the Intent-based networking system validates the intent objects. Validation including syntactic and semantic checks against network-wide policy. IBN enables instantaneous roll-back and roll-forward. If something goes wrong during a deployment push, operators only need to apply the proper versioned intent object to recover to a known good state. With well-defined policies, intent-based networking can also minimize the impact and extent of errors during the implementation of new intents. Intent-based networking supports intent-based flashback. When the system is aware of the intended results for a specific configuration, it may preserve these results despite outages or device faults by reconfiguring other network elements or employing alternative techniques to achieve the same objectives. ("What is intent-based networking," n.d.)

Intent-based networking is a declarative network operation model. It contrasts with traditional networking, which necessitates network engineers to specify the sequence of activities required on specific network devices and produces a high error risk. ("What is intent-based networking," n.d.)

Intent-Based Networking vs. Traditional Networking & Software-Defined Networking

The traditional network architecture relies on device-by-device management and requires lots of manual operations, which are inefficient and likely to cause human error. Compared to the traditional network, the intent-based network has many differences, as shown in Figure 2 below.

Figure 2

Comparison of traditional and intent-based networks

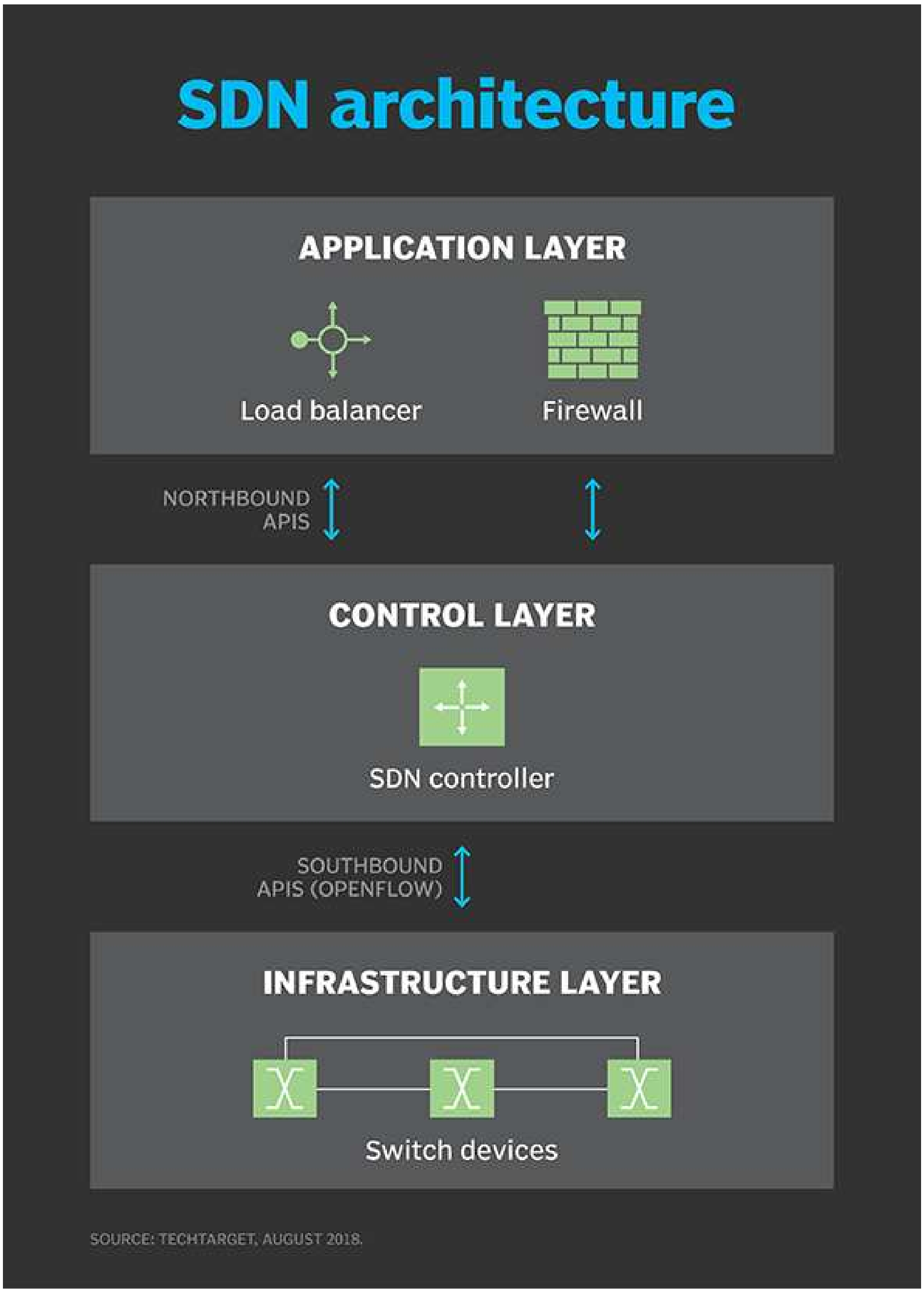
Capabilities	Traditional network	Intent-based network
Architecture	• Device-by-device management • Unidirectional configuration • Nonprogrammable devices • Patchy network security	• Networkwide system-oriented management • Closed-loop automated configuration and assurance • Programmable physical and virtualized infrastructure • Security functions integrated systematically throughout the architecture • API-centric, model-based • Open hardware and software stack
Translation	• Ad hoc operator interpretation and ad hoc translation	• Yes, through intent capturing and translation system functions
Intent verification	• No support	• Yes, integrity and consistency checks
Policy support	• Limited, expressed by device commands	• Intent-based policies based on models
Activation	• Limited (scripting), device-by-device	• Automated, networkwide with controllers
Telemetry	• Limited support	• Extensive support
Assurance	• Manual, device-by-device	• Automated, full analytics with AI/ML or formal method support
Feedback loop	• Based on ad hoc, manual operator monitoring	• Yes, automated for either operator or system activation
Outcomes	• Limited, best effort business alignment • Complex and costly to manage at scale	• Continuous business alignment • Simplified, efficient management at scale

Note. Adapted from *Cisco White Paper* (Parker, 2018, p. 7), website https://securenetworkers.com/2018/08/31/intent-based-networking/

As shown in Figure 3 below, intent-based networking and software-defined networking are different technologies, even though they are likely in many ways. Both networking models manage devices that are distributed on the network through a centralized controller rather than having to operate separate devices individually through each device's own command-line interface console as in traditional networking. Both networking models aim to decouple management from different devices and better align it with the operational requirements of businesses. Both software-defined networking and intent-based networking are capable of understanding network configuration and interaction across other devices. Due to their management controllers that comprehend the required configuration adjustment of all devices, either networking model allows an administrator to define how the network should behave, make rules, and then apply rules to a series of devices with a single command. The abstraction and networking management commands are the most significant difference between intent-based networking and software-defined networking. Intent-based networking and software-defined networking are implemented at the administrator level in separate ways. In short, though software-defined networking abstracts the device control for better management, it still has the overall view of devices. Software-defined networking issues management commands with knowledge about network parameters, such as IP address ranges, VLAN information and device details. On the other hand, intent-based networking abstracts the administrator commands of software-based networking by transforming networking management into a higher level that is business-oriented rather than device-based. After a management command (an intention) is issued in an intent-based network, the management application will make configuration changes on related devices based on the business intention. The second-level abstraction is the significant difference between intent-based networking and software-defined networking. (Fruehe, 2019)

Figure 3

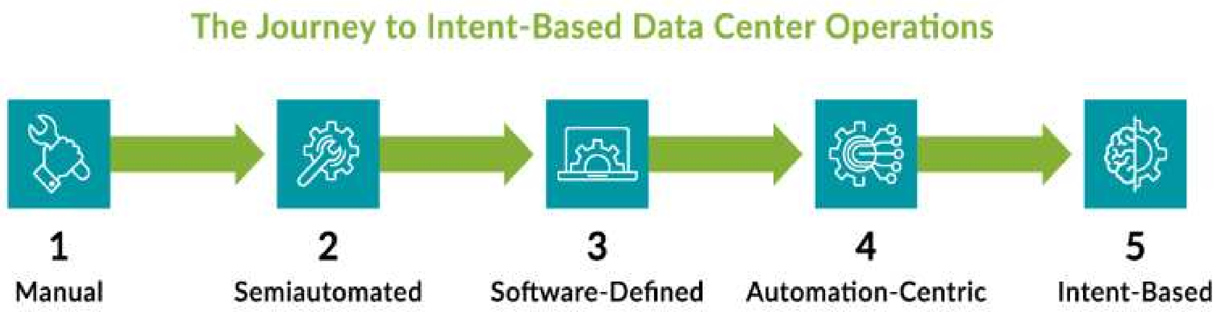
Basic Software-Defined Network Architecture



Note. Adapted from *TechTarget: Networking* (Rosencrance et al., 2022), website https://www.techtarget.com/searchnetworking/definition/software-defined-networking-SDN

Figure 4

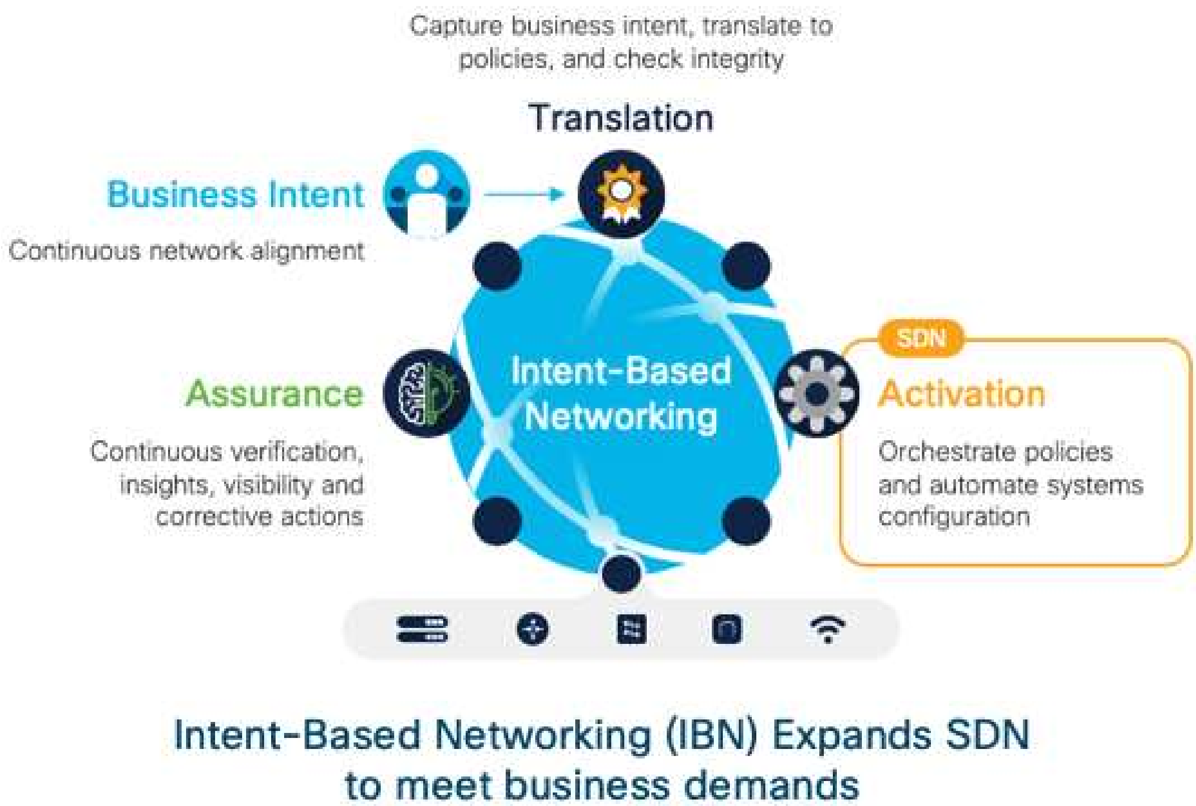
The Journey to Intent-Based Data Center Operations



Note. Adapted from *Juniper Networks*, website
https://www.juniper.net/us/en/research-topics/what-is-intent-based-networking.html

Figure 5

Relationship between Intent-Based Networking and Software-Defined Networking



Note. Adapted from *Cisco Blog*, website
https://blogs.cisco.com/networking/sdn-is-growing-up-its-called-ibn

Intent-Based Networking Architecture

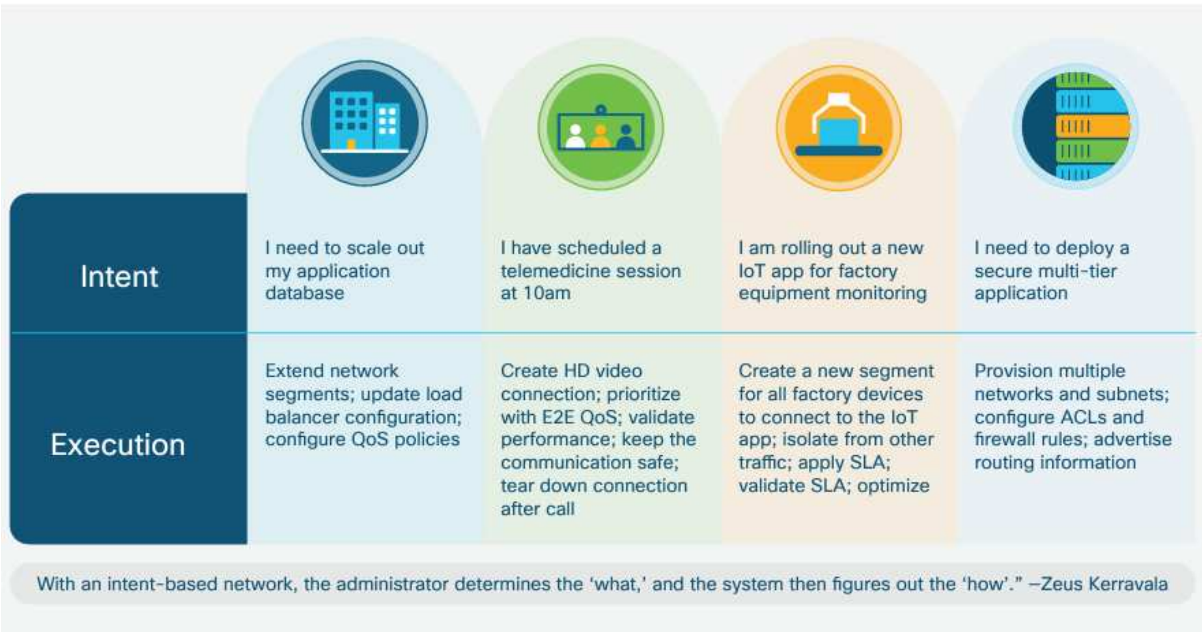
"Intent" in Context of Intent-Based Networking.

For many years, the term "intent" has been used in the telecommunications industry as an updated form of the term "policy". The concept of "intent" was created because the industry realized the difficulty of policy management, and a more straightforward solution is urgently needed by users who do not have related knowledge, software developers who are confronting many complex network interfaces, and telecommunication service providers. It indicates that intent is composed of a collection of policy types written in high-level operational and business objectives. Intents are meant to be small, simplified, explicit, and usually written by humans. So, it is necessary to "translate" intents into low-level and machine-understandable forms. The most significant purpose of developing intent-based technologies is to meet system requirements without specifying how to accomplish these goals.

In contrast to policies, intents are primarily used to enable rules to determine the behavior of a system and are regarded as a subset or type of policy. Typically, policy comprises Event-Condition-Action in network management. However, due to the abstraction of management command, intents have a more simplified structure that can also be used for network operation. Intent provides high-level operational and business goals that the network management system should achieve but does not detail the approach that the system should use to achieve the business goal. (Zeydan & Turk, 2020)

Figure 6

Examples of intent expressions



Note. Adapted from *Cisco White Paper* (Parker, 2018, p. 4), website https://securenetworkers.com/2018/08/31/intent-based-networking/

In conclusion, it can be said that intents play a role as the bridge across the gap between business and network implementation in the context of intent-based networking. At present, to achieve desired business outcomes, the cost of human force and time is high for handling the configuration of networking systems. Errors may be caused by human operation during the process of manual configuration. As the number of devices and system complexity increase, the error rate and maintenance cost of the system increase. Intent-based networks can simplify this step by automatically translating user intents into enforceable policies and providing continuous monitoring of the state of the network. To be clear, the intent that represents the business is different from the execution of the business. Intent stands for “what to do?” while execution stands for “how to get it done?”. Figure 6 above shows the differences between intent and execution.

Three Building Blocks Composing Intent-Based Network.

As shown in Figure 7, to build an intent-based network, three critical building blocks are needed: Translation, Activation and Assurance. (Parker, 2018, p. 5)

Figure 7

Building blocks of Intent-Based Networks (IBN)

Note. From *Cisco White Paper* (Parker, 2018, p. 5), website https://securenetworkers.com/2018/08/31/intent-based-networking/

Translation. In the intent-based networking model, translation requires multiple functions, and the desired intents can be characterized by using collections of operators. It can be a pre-specified syntax or language, a user-friendly graphical user interface (GUI), an abstract model such as JSON and XML, or even verbal expression in the future. This abstraction is the most significant characteristic of the intent-based network architecture, distinguishing itself from the other existing network architectures. A further responsibility of

Translation is to transfer the collected intent into a common model-based policy (MBP), typically with a controller-based architecture, so that the network nodes could react to intent correspondingly by leveraging standard MBPs.

Activation. Activation functions ensure that generated MBPs are propagated throughout all relevant network domains by the orchestration function of the intent-based network. Moreover, activation may apply other functions to generate the most optimal device configurations. The controller of a domain can correlate information about network elements, their capabilities, and their topology with the defined MBPs to configure devices appropriately. (Parker, 2018, pp. 5-6)

Assurance. Assurance has crucial responsibilities to ensure an intent-based network could work. By analyzing the context of data, assurance can validate if the desired intents have been applied correctly, and it keeps verifying if the desired intent is being achieved step by step. As shown in Figure 8 below, assurance is responsible for three features:

Continuous Verification. Assurance continuously checks on the intent-based networking system during the whole lifecycle of intent deployment. It observes all network element states and events and validates the system acting correctly at any time.

Generate Insight for Validation, Understanding, and Prediction. This behavior requires support from artificial intelligence and machine learning technologies. It is possible to evaluate the feasibility of the intent before trying to achieve it and predict and validate the stability and performance of the network system.

Conduct Corrective Action and Optimization. The intent-based network allows a mechanism for automating the correction of any intent-based policy infractions or for automating continual network optimizations to ensure that the declared intent is always realized. Assurance leverages a closed-loop cycle to support this behavior. (Parker, 2018, pp. 5-6)

Figure 8

Three main aspects of Assurance

Note. Adapted from *Cisco White Paper* (Parker, 2018, p. 6), website https://securenetworkers.com/2018/08/31/intent-based-networking/

Artificial Intelligence Involved in Intent-Based Networking.

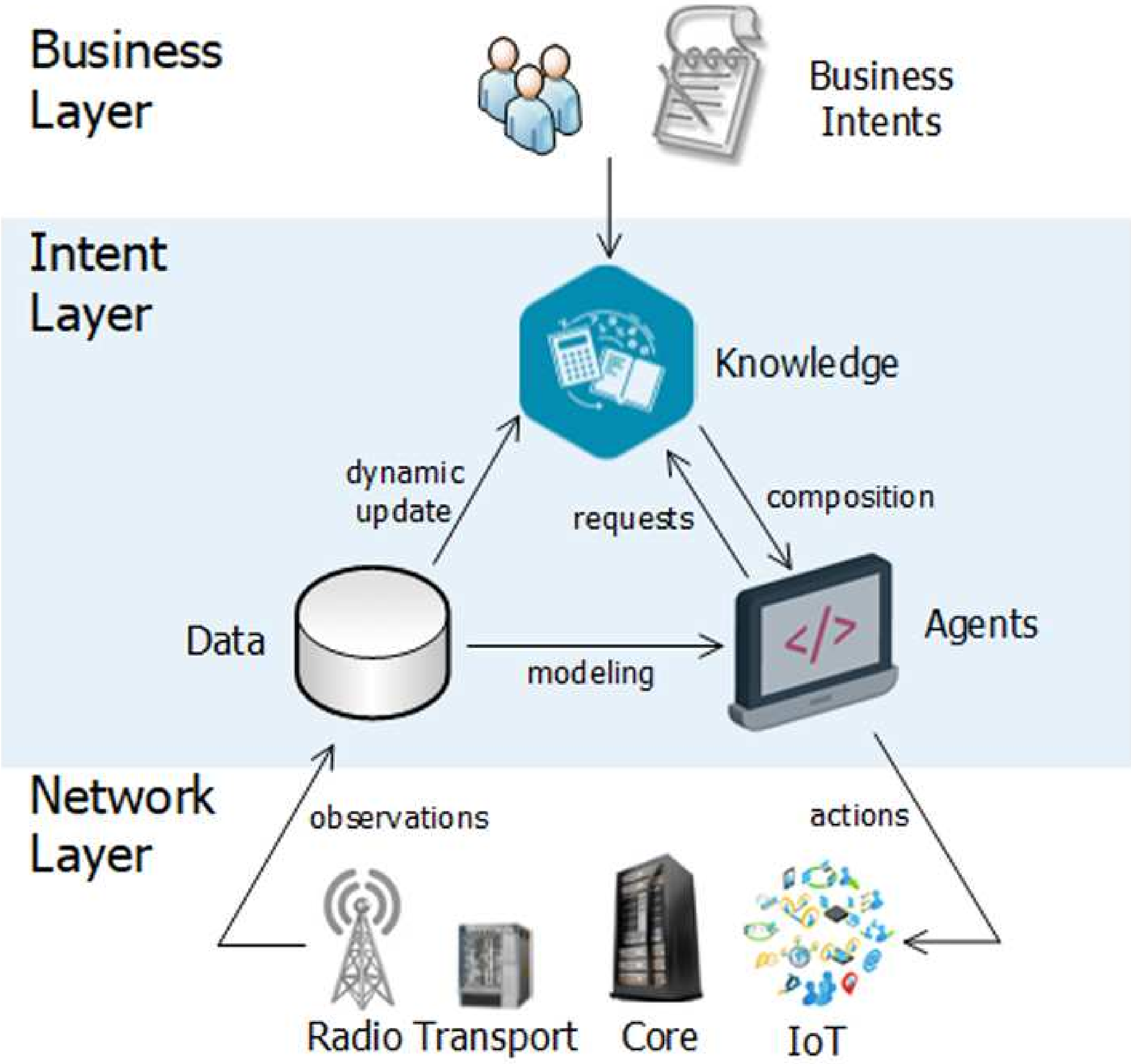
To deal with business intents typically written in a minor, simplified, and high-level language, intent-based networking relies on artificial intelligence. Specifically, intent-based networking needs AI technology like Natural Language Processing to help networking management systems understand the meaning of intent made by the upper layer. Natural Language Understanding systems can structure the expression in a specific language for further automated analysis and execution. Artificial intelligence technologies help abstract network management services, effectively reducing the cost and complexity of administration and maintenance while improving efficiency and fault tolerance. The significant problems with the intent-driven networking management model can be solved with the development of AI and deep learning technologies. In other words, artificial intelligence is the cornerstone of intent-based networking development. It is a critical feature that distinguishes intent-based networking from software-defined networking and traditional networking.

An Illustrative Architecture of Intent-Based Networking.

Currently, technologies and theories related to intent-based networking are still under development and await further research and enhancement. Combining emerging technologies, such as AI, with the existing network management system enables the creation of one possible form of intention-based networking in Figure 9.

Figure 9

Illustrative intent-based networking architecture



Note. Adapted from *Recent Advances in Intent-Based Networking: A Survey* (Zeydan & Turk, 2020, p.2), doi: 10.1109/VTC2020-Spring48590.2020.9128422

To solve the problems the traditional networks encounter and develop more network management features, automation to handling intents is involved in this architecture. This illustrative intent-based networking architecture is a three-layered architecture comprising business layer, intent layer, and network layer. The three layers each have different responsibilities.

Business Layer. It is where business intents are initiated. All business intents are provided by users and specify basic information of intents such as business goals, processes, and targets. The intent layer's purpose is not to mindlessly perform the entire intended sequence of operations but rather to re-evaluate and re-plan after each step. There is the possibility that multiple business intents conflict with each other. Detecting and solving these kinds of conflicts is the responsibility of the business layer.

Intent Layer. In the intent layer, there are three essential components: knowledge, agent, and data. The knowledge component is responsible for dealing with the abstraction of intents. It initiates the analysis of intents and then sends the composition of the analysis outcome to the agent component. This layer's agent contains the network objects' interface and is responsible for initiating actions on specific network devices or objects. Before the agent starts actions on network objects, it needs to send requests to the knowledge for reasoning because it does not have its own intelligence component. The third component in this layer, data, contains network topology and inventory information. It observes all network changes and then synchronizes those changes to the knowledge in real time. The data model the network topology and transmits the model to the Agent.

Network Layer. The last layer is the network layer, which includes all physical network devices known as network objects. It is responsible for holding the hardware abstraction model and carrying out actions that the agent requests from the intent layer. The network layer is capable of translating low-level network data into formal representation to help the intent layer work with it efficiently. (Zeydan & Turk, 2020, p. 1)

The other characteristics of each layer are shown in Table 1 (Zeydan & Turk, 2020, p. 4).

Table 1

Characteristics of Layers in Intent-Based Networking Architecture

Layer		**Characteristics**
Business Layer		— Higher-level declarative policy that operates at the level of a network and services — Provide semantics to consume network resources —Allowing high-level guidance by a central entity —Detect and resolve conflicts between multiple intents
Intent Layer	**Knowledge**	— Access to knowledge and execute judgement —Performs inference from relations between objects
	Agent	— Capture the business intent and translate into policies — Utilize ontology-based approach to communicate with users —Communication interface directly to the network objects
	Data	— Keep the state of each intent and the relation between network objects — Provides models for the observed data — Provides algorithms for data modelling

Network Layer	— Present the abstraction of domain-specific data and control plane technologies — Specify context-aware architecture for enhancing the network intelligence

Note. Adapted from *Recent Advances in Intent-Based Networking: A Survey* (Zeydan & Turk, 2020, p.4), doi: 10.1109/VTC2020-Spring48590.2020.9128422

Lifecycle of Intent

The intent is subject to a lifecycle: after it is generated, it may undergo modifications throughout time and may be withdrawn at some point. This lifespan is tightly correlated with the interconnection functions associated with the intent. A general lifecycle of intent and its primary functions are shown in Figure 10. Figure 10 shows two functions related to intent: one is responsible for fulfillment, and another is for assurance.

Fulfillment. Fulfillment is related to the functions that take an intent from its creation by an administrator to its fulfillment. This consists of the following:

Interaction Functions. Functions that recognize intent based on interaction with users and functions that allow users to refine their purpose and communicate it in such a way that an Intent-Based System may act on it. These functions may include unorthodox human-machine interactions, in which a human will not merely provide simple commands but instead engage in a dialogue with the computer to provide clarifications, explain repercussions and trade-offs, and facilitate modifications (Clemm et al., 2020).

Translation Functions. It is the function that transforms user intent into network-relevant actions and requests. This function involves deep-learning algorithms for getting optimal results by deciding the best actions to achieve it.

Execution Functions. After the translation functions confirm the actions that need to be done, execution functions are responsible for executing and orchestrating the actions.

Assurance. Assurance includes functions that check if the desired intent is achieved after the action is executed. This includes:

Monitoring and Observation Functions. The monitoring and observation functions are for the network and its states and behaviors.

Assessment and Validation Functions. Functions can assess and validate whether the outcome conforms to the intent.

Corrective Functions. The Functions can respond to errors with corrective actions if needed.

Abstraction Functions. These functions can abstract the analytic outcomes and results gained by observation functions in a user-friendly form.

Presentation Function. These functions are responsible for presenting the summarization and visualization of the intent lifecycle. This includes the function reporting the intent compliance status. (Clemm et al., 2020)

There are also three vertical spaces shown in Figure 10. The user space makes the connections between users and the intent-based network system. The translation/IBS space handles the transformation between high-level user intents and low-level actions the network system takes. The network operation space provides functions that deal with the traditional low-level configuration to achieve user intents.

Figure 10 also shows that there are two different loops in the lifecycle of intent. The “inner” loop is an intent processing loop in translation / IBS and network operation spaces. The inner loop is fully automated, and no human operation is required. The inner loop is a constant process of taking action, observing, verifying, and adjusting until the intent is achieved. The “outer loop involves all spaces, including the user space. With human intervention, intent can be adjusted based on information gained from the IBS space. (Clemm et al., 2020)

This intent lifecycle is not fixed. Figure 11 shows an example of an alternative intent lifecycle with a separate flow between user space and network space so that the feedback gained from observation is reported directly to the user space from the network space.

Figure 10

Intent Lifecycle

```
           User Space   :       Translation / IBS        :   Network Ops
                        :             Space              :      Space
                        :                                :
          +---------+   :  +----------+  +-----------+   :  +---------+
 Fulfill  |recognize| ---> |translate/|-->|learn/plan/| ---> | config/ |
          |intent   | <--- |  refine  |  |  render   |   :  |provision|
          +----^----+   :  +----------+  +-----^-----+   :  +---------+
               |        :                      |         :       |
 ..............|...............................|.................|.....
               |        :                 +----+---+     :       v
               |        :                 |validate|     :  +----------+
               |        :                 +----^---+ <------| monitor/ |
 Assure     +---+---+   :  +---------+   +-----+---+     :  | observe/ |
            |report | <--- |abstract |<---| analyze | <------|  assure  |
            +-------+   :  +---------+   |aggregate|     :  +----------+
                        :                +---------+     :
```

Note. Adapted from *Intent-based networking - concepts and Overview* (Clemm et al., 2020), website https://datatracker.ietf.org/doc/draft-clemm-nmrg-dist-intent/

Figure 11

Altered Intent Lifecycle

```
         user related                   user data <-----<-+--------+
         data  +                          +               |        |
               |                          |               |        |
          +----v------+               +-----v-----+       |        |
          | recognize +---+    +-----+ generate  |        |        |
    user  +-----------+   |    |      +-----------+       |        |
    space                 |    |                          |        |
 +-----------------------------------------------------------------+
    system                |    |                          |        |
    space            +---v---v---+   +----------+   +-----+-----+  |
                     | translate <-->+ validate <---> recommend |  |
                     +-----+-----+   +----------+   +-----------+  |
                           |                                       |
                     +-----v-----+                                 |
                     | normalize |                                 |
                     +-----+-----+                                 |
                           |                                       |
                     +-----v-----+                                 |
                     | decompose |                                 |
                     +-----+-----+                                 |
                           |                                       |
                    +------v------+                                |
                    | communicate |                                |
                    +------+------+                                |
     preparation           |                                       |
     phase                 |                                       |
  +----------------------------------------------------------------+
     operation             |                                       |
     phase           +-----v----+                                  |
                     | fullfill |                                  |
                     +-----+----+                                  |
                           |                                       |
                      +----v----+     +--------+                   |
                      | observe +-----> report +-------------------+
                      +----+----+     +--------+
                           |
                      +----v---+
                      | assure |
                      +--------+
```

Note. Adapted from *Intent-based networking - concepts and Overview* (Clemm et al., 2020), website https://datatracker.ietf.org/doc/draft-clemm-nmrg-dist-intent/

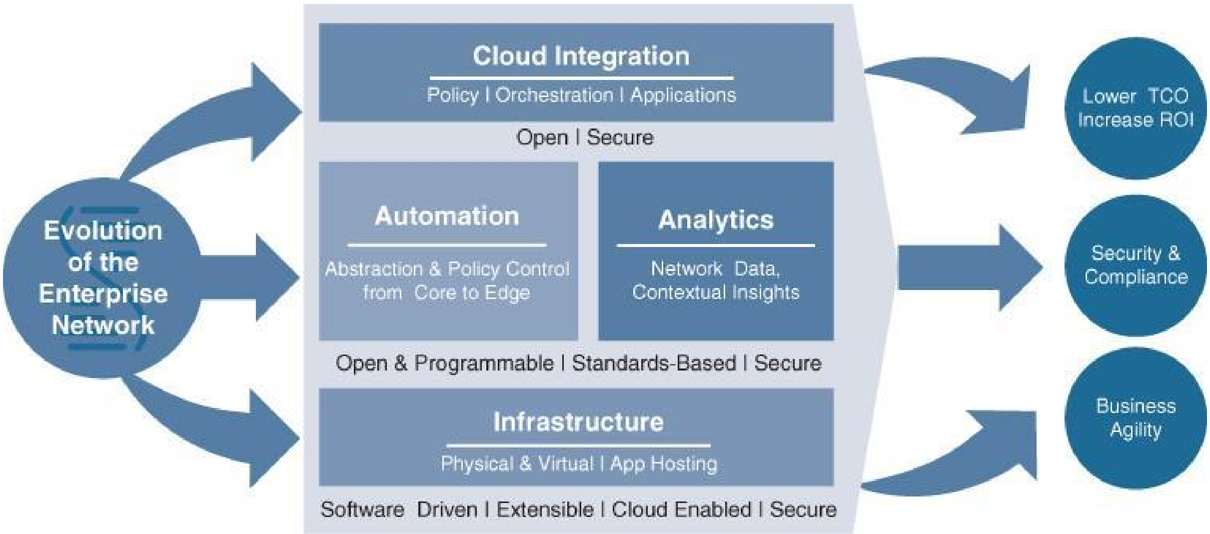
Application of Intent-Based Networking

Cisco Digital Network Architecture: Intent-Based Networking for Enterprise

With the increasing complexity of enterprise networks, the complexity and costs associated with maintenance and management are skyrocketing. Cisco proposed the Cisco Digital Network Architecture (Cisco DNA) leveraging intent-based networking to help the digital transformations of enterprise network systems to the next generation.

Figure 12

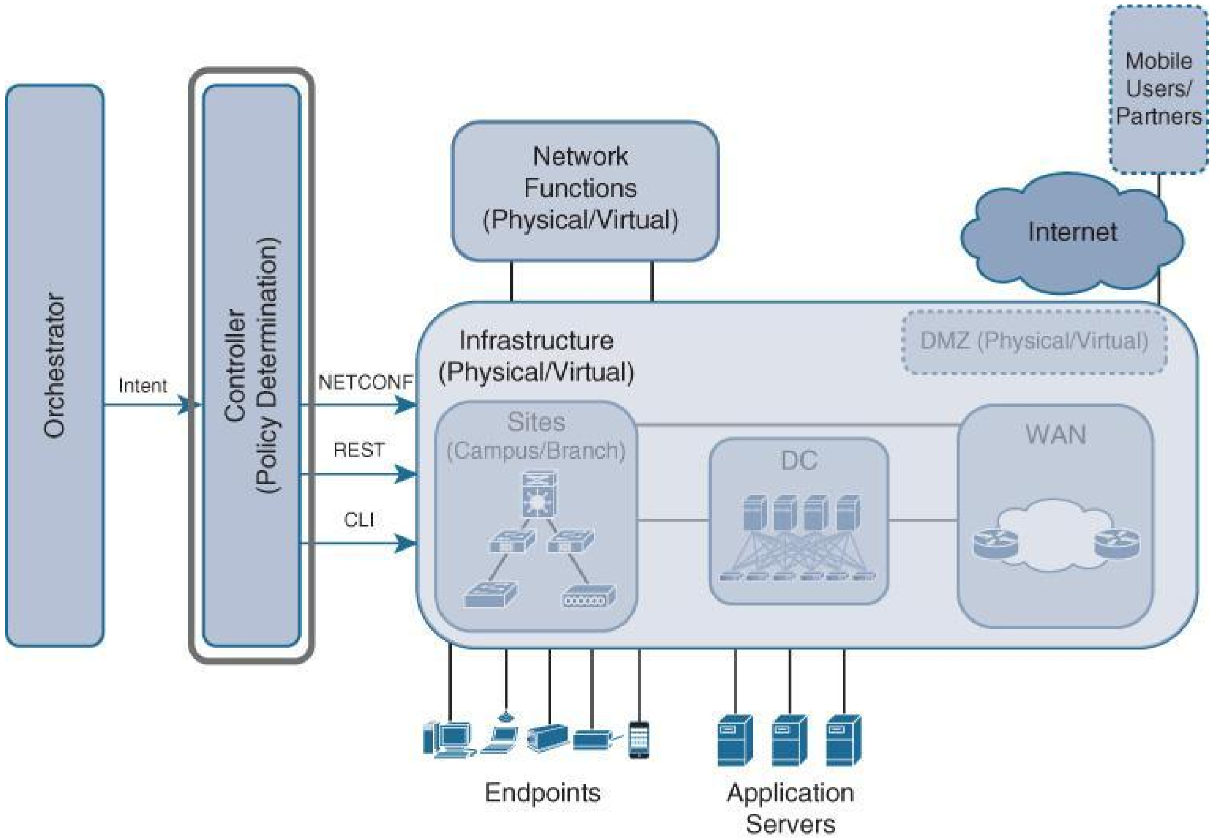
Cisco DNA Concepts and Main Components



Note. Adapted from *Cisco DNA Concepts and Main Components* (Szigeti et al., 2019), website https://learning.oreilly.com/library/view/cisco-digital-network/9780134723952/?sso_link=yes&sso_link_from=university-of-alberta

Figure 13

Cisco DNA Controllers (Network Abstraction and Policy Determination)



Note. Adapted from *Cisco DNA Concepts and Main Components* (Szigeti et al., 2019), website https://learning.oreilly.com/library/view/cisco-digital-network/9780134723952/?sso_link=yes&sso_link_from=university-of-alberta

As concepts are shown in Figure 12, Cisco DNA abstract the entire network organization to provide a programmable network system on which the network administrators can operate the entire network. It dramatically reduces the time cost and labor cost of device management, network configuration and other maintenance, and strengthens the scalability of enterprise networks. The components of Cisco DNA architecture include:

- A low-level network with devices
- A logical overlay topology
- A controller (Figure 13)

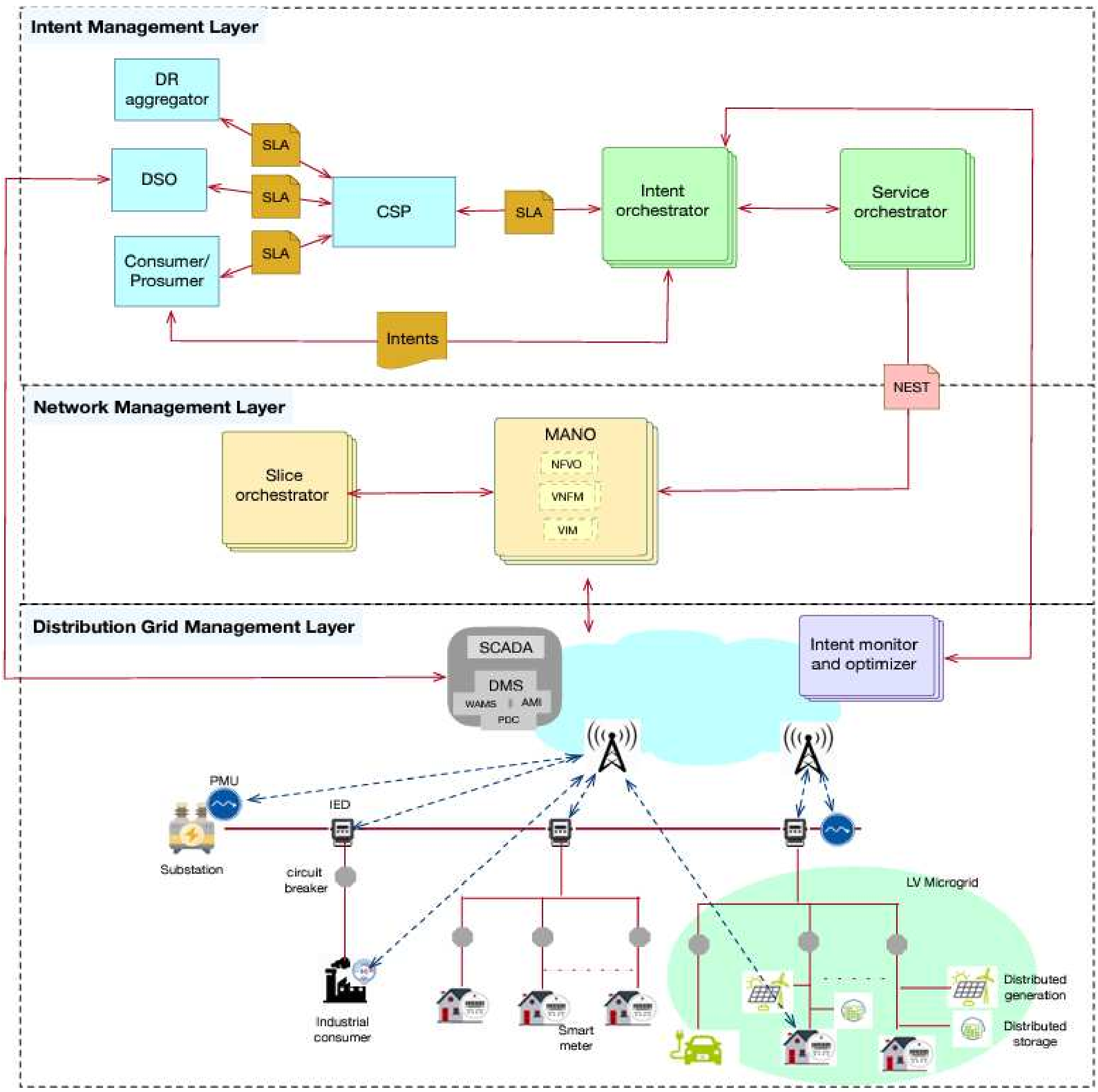
The controller is responsible for abstraction and intent translation. It transforms business intents into machine-understandable outcomes and automates actions to achieve them. It also provides analysis features for the optimization of the network capacity and more efficient troubleshooting. Cisco DNA can promote the digital transformation of enterprises and help enterprises cater to future demands. (Szigeti et al., 2019)

Intent-Based Network Management for Smart Distribution Grids

Another application case of the intent-based network for smart distribution grids is proposed. Automation is essential for the management and operation of distribution grid systems. By using intent-based networking, it is possible to build a service-oriented distribution grid system that allows users to operate the network on a high-level system of abstraction. The automated "inner" loop of intent-based networking realizes the automation part of the distribution grids.

Figure 14

Proposed Intent-based network management and service orchestration architecture



Note. Adapted from *Intent-based Network Management and Orchestration for Smart Distribution Grids* (Mehmood et al., 2021), 2021 28th International Conference on Telecommunications (ICT), 1-6.

As shown in Figure 14, the intent layer is crucial to processing. The intent is initially passed from the consumer/prosumer to the intent orchestrator. After the intent transformation, optimization, and validation with the intent monitor and optimizer module, the intent orchestrator forwards service requirements to the service orchestrator. The service orchestrator generates service models and forwards them to the network management layer. (Mehmood et al., 2021)

Conclusion

In this report, some aspects of intent-based networks are analyzed and studied. I first briefly introduced traditional and software-defined networks, then comprehensively analyzed the advantages of intent-based networks on this basis. I then focused on the analysis of intent-based networks, including the definition of intents in the context, the components of intent-based networks, associated artificial intelligence technologies, the lifecycle of intents within an intent-based networking system, and an Illustrative intent-based network architecture. Finally, two applicable cases of IBN are mentioned and introduced. This analysis report of the intent-based network indicates the intent-based networks' current design motivation and general architecture. With intent-based networks, network maintenance costs can be dramatically reduced while network scalability and management efficiency get greatly enhanced. It can also be expected that, with the development of artificial intelligence and deep learning technologies related to language analysis, the concepts of intent-based networks will be further improved in the direction of user-friendliness and administrative ease.

List of Figures

List of Tables

References

Bose, M. (2022, March 28). *Types of network topology explained.* Official NAKIVO Blog. Retrieved March 3, 2023, from https://www.nakivo.com/blog/types-of-network-topology-explained/

Clemm, A., Ciavaglia, L., Granville, L. Z., & Tantsura, J. (2020, January 14*). Intent-based networking - concepts and Overview.* IETF Datatracker. Retrieved March 5, 2023, from https://datatracker.ietf.org/doc/draft-clemm-nmrg-dist-intent/

E. Zeydan and Y. Turk, *Recent Advances in Intent-Based Networking: A Survey,* 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring), Antwerp, Belgium, 2020, pp. 1–5, doi: 10.1109/VTC2020-Spring48590.2020.9128422.

Fruehe, J. (2019, January 17). *Intent-based networking vs. SDN: How do they differ?* Networking. Retrieved February 27, 2023, from https://www.techtarget.com/searchnetworking/answer/Intent-based-networking-vs-SDN-How-do-they-differ

GeeksforGeeks. (2020, August 21). *Difference between software-defined network and Traditional Network*. GeeksforGeeks. Retrieved February 27, 2023, from https://www.geeksforgeeks.org/difference-between-software-defined-network-and-traditional-network/

Gilad, J. (2020, October 12). *SDN is growing up – it's called IBN*. Cisco Blogs. Retrieved February 27, 2023, from https://blogs.cisco.com/networking/sdn-is-growing-up-its-called-ibn

Mehmood, K., Mendis, H.V., Kralevska, K., & Heegaard, P.E. (2021). *Intent-based Network Management and Orchestration for Smart Distribution Grids*. 2021 28th International Conference on Telecommunications (ICT), 1-6.

Parker, K. (2018, August 31). *Cisco Systems - Intent-based networking: Building the bridge between business and it.* Secure Networkers. Retrieved February 28, 2023, from https://securenetworkers.com/2018/08/31/intent-based-networking/

Rosencrance, L., English, J., & Burke, J. (2022, May 11). *What is software-defined networking (SDN)?* definition from techtarget.com. Networking. Retrieved March 3, 2023, from https://www.techtarget.com/searchnetworking/definition/software-defined-networking-SDN

Software-defined networking (SDN) Vs Traditional Networking. Kyndryl. (n.d.). Retrieved February 24, 2023, from https://www.kyndryl.com/ca/en/learn/sdn-vs-traditional-networking

Szigeti, T., Zacks, D., Falkner, M., & Arena, S. (2019). *Cisco digital network architecture : intent-based networking for the enterprise*. Cisco Press. Retrieved March 12, 2023, from https://learning.oreilly.com/library/view/cisco-digital-network/9780134723952/?sso_link=yes&sso_link_from=university-of-alberta

What is intent-based networking?: VMware glossary. VMware. (n.d.). Retrieved February 17, 2023, from https://www.vmware.com/topics/glossary/content/intent-based-networking.html?resource=cat-509765043#cat-509765043

What is intent-based networking?. Juniper Networks US. (n.d.). Retrieved February 21, 2023, from https://www.juniper.net/us/en/research-topics/what-is-intent-based-networking.html