Journal of IoT in Social, Mobile, Analytics, and Cloud (ISSN: 2582-1369)
www.irojournals.com/iroismac/

Agent-Based Computing for Autonomic Networking: A Comprehensive Survey

Ranganathan G.

Principal, Dhaanish Ahmed Institute of Technology, Coimbatore, India.

E-mail: ranganathang.samy@gmail.com

Abstract

The rapid growth of large-scale, heterogeneous, and highly dynamic networked systems has rendered traditional, manual, and centrally managed network control mechanisms less effective. Autonomic networking has been identified as an emerging technology that will allow networked systems to manage themselves through self-configuration, self-optimization, self-healing, and self-protection capabilities. The agent-based computing paradigm appears to be well suited to support autonomic networking through decentralized, adaptive, and intelligent decision-making capabilities. This paper provides a comprehensive survey of agent-based computing paradigms for autonomic networking, including existing architectures, models, and coordination paradigms. The survey of existing works will be categorized based on agent responsibilities, control paradigms, and learning capabilities. Additionally, the survey will provide a comparative analysis of existing architectural paradigms based on latency, scalability, coordination overhead, and convergence properties. Moreover, the paper will discuss the challenges and open issues of agent-based autonomic networking paradigms, including scalability, coordination overhead, security, and trustworthiness. Future research directions will be identified to develop fully autonomous networked systems.

Keywords: Autonomic Networking, Agent-based Computing, Multi-Agent Systems, Self-Managing Networks, Intelligent Network Control.

1. Introduction

Modern-day network infrastructures have advanced beyond traditional static communications systems. The widespread adoption of cloud computing, Software-Defined Networking (SDN), the Internet of Things (IoT), and novel 5G and 6G technologies has made

DOI: https://doi.org/10.36548/jismac.2026.1.003
Received: 18.01.2026, received in revised form: 16.02.2026, accepted: 07.03.2026, published: 16.03.2026

modern-day network infrastructures highly dynamic and increasingly difficult to manage due to their complexity. Traditional manual management and centralized control techniques have shown their inability to manage such dynamic and complex network infrastructures, which often lead to increased delays and decreased reliability in network operations. Thus, to manage such dynamic and complex network infrastructures, autonomic networking was conceived as a novel paradigm to provide these infrastructures with self-managing capabilities using minimal human intervention. Autonomic networking is inspired by nature and aims to provide network infrastructures with self-managing capabilities such as self-configuration, self-optimization, self-healing, and self-protection. These autonomic characteristics in network infrastructures can be achieved using distributed intelligence and adaptability, which are difficult to attain in traditional centralized architectures. In this regard, agent-based computing has attracted significant attention as a novel computational paradigm for autonomic networking infrastructures.

1.1 Background

1.1.1 Autonomic Computing and Networking

Autonomic networking refers to a networking paradigm whereby network infrastructures have the capability to manage themselves with little or no need for human intervention. The autonomic networking concept is based on the autonomic computing concept, whereby autonomic networks have the capability for self-management, self-configuration, self-optimization, self-healing, and self-protection, enabling network infrastructure to adapt dynamically to changing conditions. The autonomic computing model is usually described by the Monitor-Analyze-Plan-Execute over a shared Knowledge base (MAPE-K) loop, which is used for self-management. The autonomic networking concept, therefore, refers to the extension of the autonomic computing concept to networking, whereby network infrastructure dynamically changes to adapt to changing conditions without external control. Figure 1 conceptually illustrates the motivation behind autonomic computing. As infrastructures become larger in scale, more heterogeneous, and more complex in operation, managing them manually becomes inefficient and prone to errors. In this context, the concept of autonomic computing has been introduced, providing the ability to manage computing infrastructure in a self-managing way. In other words, the computing infrastructure can monitor its conditions, analyze its behavior, and take corrective actions to improve its operation. Similarly, the concept of autonomic computing has been used to develop autonomic networking, where the same principles of self-management are applied to network infrastructures. In other words, autonomic networking has been designed to improve the operation of the network infrastructures in terms of cost, reliability, and responsiveness.

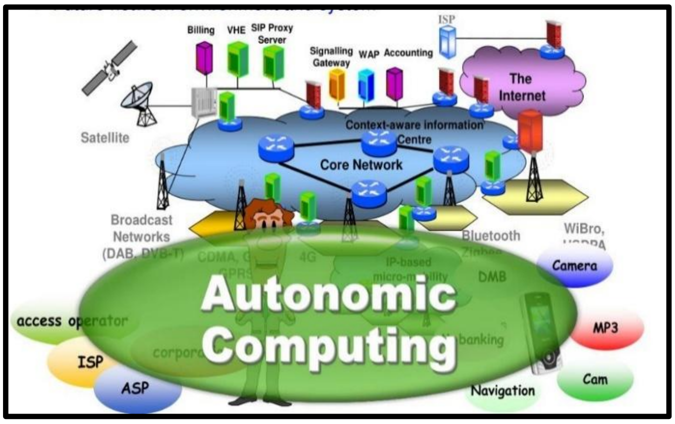

Figure 1. The Need for Autonomic Computing [16]

1.1.2 Agent-Based Computing

Agent-based computing offers an abstraction that is naturally suited to modeling autonomous and intelligent agents in distributed environments. An agent is defined as an autonomous entity that can perceive the environment, make decisions, and perform actions to accomplish particular goals. In MAS, multiple agents interact, cooperate, or compete to solve complex problems that cannot be easily solved by central control.

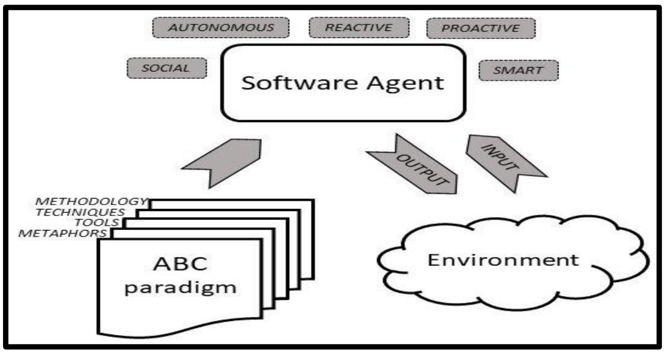

Figure 2. Agent-Based Computing (ABC) Paradigm Illustrating the Interaction Between a Software Agent and Its Environment [17]

Fig. 2 depicts the basic paradigm of agent-based computing, where an autonomous software agent interacts with the environment by perceiving and acting. An agent continually

ISSN: 2582-1369

senses environmental conditions, processes the perceived information by utilizing internal reasoning capabilities, and takes actions that affect the environment. This perception, decision, and action cycle allows agents to respond to changing conditions and reach particular goals. In networking environments, agents can be used to represent network nodes, controllers, or management components that can autonomously perceive network states and take actions to manage the network.

1.2 Motivation for Agent-Based Autonomic Networking

The motivation for applying agent-based computing to autonomic networking arises from the inherent limitations of traditional network control models. Centralized control mechanisms often suffer from scalability bottlenecks, single points of failure, and limited adaptability to rapid environmental changes. In contrast, agent-based approaches support distributed intelligence, allowing network entities to make local decisions based on real-time observations while collectively contributing to global objectives. Agent-based autonomic networking enables networks to respond more effectively to dynamic conditions such as traffic fluctuations, component failures, and changing service demands. The ability of agents to learn, cooperate, and negotiate further enhances their suitability for complex networking scenarios, where static policies and predefined rules are insufficient. Consequently, agent-based computing has been increasingly adopted in autonomic network architectures for tasks such as resource allocation, fault management, performance optimization, and adaptive routing across diverse networking environments.

1.3 Practical Scenarios in Agent-Based Autonomic Networking

Case Study 1: Resilient SDN Control in Geographically Distributed Networks

SDN architectures traditionally rely on logically centralized controllers that maintain a global view of network topology and traffic conditions. While centralization simplifies global optimization and policy enforcement, it introduces critical scalability and reliability challenges in large-scale or geographically distributed deployments. Consider a multi-domain cloud data center infrastructure interconnected across multiple regions. During peak traffic hours or sudden traffic bursts caused by large-scale service requests, the centralized SDN controller may become overloaded. Additionally, controller failures or communication link disruptions can delay routing updates and congestion mitigation actions.

In an agent-based autonomic networking architecture, decentralized control agents are deployed at domain controllers or edge switches. These agents continuously monitor local traffic conditions and independently execute localized MAPE-K control loops. Upon detecting congestion or controller unavailability, local agents can immediately initiate rerouting decisions, traffic shaping, or load redistribution without waiting for centralized instructions. Coordination among agents ensures consistency with global policies while preserving rapid local responsiveness. This distributed decision-making significantly reduces recovery time, mitigates single points of failure, and enhances scalability. The case demonstrates how agent-based architectures directly address latency, fault tolerance, and scalability limitations inherent in centralized SDN systems.

Case Study 2: Adaptive Resource Management in Large-Scale IoT Deployments

Internet of Things (IoT) systems have thousands of heterogeneous devices, each operating under limited energy, bandwidth, and computing resources. The network traffic in IoT systems is very dynamic due to the mobility of the devices, intermittent connectivity, environmental interference, and varying workload. In a centralized management system, network configuration policies are fixed and updated at regular intervals using a centralized management server. However, such fixed policies cannot handle the dynamic changes in the environment. For instance, in a smart city environment, if the network is suddenly congested due to an increase in sensor devices, the quality of the gathered information might be affected, leading to an increase in transmission delays. In an agent-based autonomic networking architecture, the gateway agents continuously observe the activities, energy, and network load of the IoT devices. The learning agents can dynamically adjust the routing paths, transmission intervals, and power management strategies according to the network traffic. The agents using reinforcement learning can optimize network performance by choosing the most suitable channel for transmission while conserving energy. The autonomic networking system enables the gateway agents to cooperate, leading to an improvement in network performance, energy efficiency, and fault tolerance.

Case Study 3: Stability and Safety in Learning-Driven Autonomous Networks

Emerging 5G/6G and edge computing networks increasingly incorporate intelligent traffic optimization and autonomous service orchestration. However, incorporating such intelligent agents into live networks often brings about issues regarding their stability and

ISSN: 2582-1369

safety. For example, when incorporating reinforcement learning-based intelligent agents for optimizing traffic routes, these agents might initially explore suboptimal routes. This could lead to a certain level of congestion in the network. In safety-critical environments such as vehicular networks or industrial automation systems, these instabilities might prove to be critical. Agent-based autonomic networking addresses this very issue by incorporating a hybrid framework for agent-based autonomic networking. Baseline policies ensure safety constraints, while intelligent agents ensure optimized performance within certain limits. Thus, this case highlights the importance of incorporating intelligent agents with guarantees for their stability.

Across these representative case studies, a consistent pattern emerges: centralized architectures face scalability and recovery limitations, static rule-based systems lack adaptability, and purely learning-driven systems introduce stability risks. Agent-based autonomic networking provides a balanced architectural solution by enabling distributed intelligence, localized adaptation, cooperative coordination, and hybrid learning-control mechanisms. These practical scenarios reinforce the necessity of agent-based approaches for next-generation autonomous network infrastructures.

1.4 Contributions of this Survey

1. It provides a comprehensive and network-centric review of agent-based computing approaches for autonomic networking, covering architectures, models, and coordination mechanisms
2. It presents a structured taxonomy that classifies existing work based on agent roles, control strategies, and learning capabilities
3. It critically analyzes application domains such as SDN, cloud and edge computing, and IoT networks, highlighting the strengths and limitations of agent-based solutions
4. It identifies key challenges and open research issues related to scalability, coordination overhead, security, and trustworthiness in agent-based autonomic networking
5. It outlines future research directions to guide the development of intelligent, fully autonomous network systems.

2. Literature Review

Research on autonomic networking and agent-based computing has evolved across multiple disciplines, including distributed systems, network management, multi-agent systems, and intelligent control. Recent advances in autonomous networking and agentic-based AI architectures have further expanded the scope of intelligent network automation and self-managing infrastructures [18], [19], [20]. Early studies established the foundational principles of agent-based computing and multi-agent systems, highlighting their suitability for modeling complex, decentralized, and adaptive systems [1], [6], [11]. These works emphasized core agent properties such as autonomy, reactivity, proactiveness, and social interaction, which later became central to autonomic network design. Studies on autonomic computing provided a complete overview of self-managing system concepts, including self-configuration, self-optimization, self-healing, and self-protection [3], [12], [13]. These studies explored autonomic architectures, control loops, and evaluation criteria but largely treated networking as one of many application domains rather than a primary focus. As a result, the networking-specific implications of decentralized decision-making and coordination were not deeply examined.

Several works investigated the role of agent-based approaches in achieving autonomic behavior through decentralized control and adaptive modelling [9], [14]. These studies demonstrated that agent-based models can effectively capture local interactions and global emergent behavior, making them attractive for managing dynamic network environments. However, such efforts primarily focused on conceptual modelling and theoretical analysis, offering limited insight into large-scale deployment challenges. More recent research has explored agent-based architectures for network automation and autonomic network management [4], [5]. These contributions proposed multi-agent frameworks capable of monitoring, analyzing, and optimizing network performance in a distributed manner. While these architectures demonstrated more flexibility and robustness than centralized systems, their evaluations were frequently restricted to specific scenarios, limiting their adaptability across heterogeneous network environments.

The application of agent-based computing has also been studied in related domains such as cloud computing, the Internet of Things, and autonomous systems [2], [10], [15]. These surveys highlighted the effectiveness of agents in handling scalability, dynamic resource allocation, and decentralized coordination. Nevertheless, most of these works addressed domain-specific challenges and did not explicitly integrate autonomic networking principles with agent-based control mechanisms. Recent surveys on agentic AI have further expanded the discussion by introducing goal-driven, self-directed intelligent agents capable of complex reasoning and long-term autonomy [7], [8], [18], [19]. While these studies offer valuable insights into next-generation agent architectures, their focus remains largely conceptual and application-agnostic, with limited attention to networking-specific constraints.

Table 1. Summary of Existing Works and Identified Research Gaps in Agent-Based Autonomic Networking

Research Focus	Models Used	Major Contributions and Findings	Identified Gap	References
Foundations of agent-based computing and agent-based models	Agent-Based Models (ABM), Multi-Agent Systems (MAS)	Established core agent concepts, visual taxonomies, and modelling paradigms for complex adaptive systems	Limited focus on networking-specific autonomic behaviour	[1], [6], [11]
Autonomic computing principles and architectures	MAPE-K loop, rule-based control, decentralized management	Defined autonomic properties, architectural models, and evaluation criteria for self-managing systems	Networking treated as a secondary application domain	[3], [12], [13]
Agent-based modelling for decentralized and autonomic control	Agent-based modelling, decentralized control mechanisms	Demonstrated suitability of agents for adaptive and decentralized system control	Lacked large-scale network deployment validation	[9], [14]
Agent-based autonomic network management	Multi-agent architectures, cooperative agents	Proposed decentralized network management architectures with improved adaptability and resilience	Evaluations limited to specific scenarios and network settings	[4], [5]
Agent-based computing in cloud and	Mobile agents, service agents	Highlighted agent effectiveness for scalability, resource	Weak integration with autonomic	[10]

distributed systems		management, and dynamic adaptation	networking frameworks	
Agent-based approaches in autonomous and cyber-physical systems	Agent-based simulation, MARL	Demonstrated decentralized decision-making in dynamic and safety-critical environments	Networking constraints not explicitly addressed	[2], [15]
Agentic AI and next-generation autonomous agents	Goal-driven agents, autonomous AI systems	Introduced advanced agent reasoning, long-term autonomy, and adaptive goal management	Lack of networking-centric evaluation and deployment analysis	[7], [8], [18], [19]

3. Taxonomy of Agent-Based Autonomic Networking

To systematically examine studies on agent-based autonomic networking, a structured taxonomy is required to comprehend the design decisions, operational characteristics, and limits of existing systems. Agent-based autonomic networking solutions vary significantly in how intelligence is distributed, how decisions are made, and how agents coordinate to achieve self-managing behaviour. Based on the reviewed literature, these solutions can be classified along several key dimensions, including agent roles, control strategies, learning capabilities, network scope, and coordination mechanisms [1], [3], [11], [12]. Fig. 3 presents a conceptual taxonomy of agent-based autonomic networking that organizes existing research according to several key design dimensions. These dimensions include agent roles, control strategies, learning capabilities, network scope, and coordination mechanisms. The taxonomy highlights how different architectural and methodological choices influence system behaviour, scalability, and adaptability. By structuring the literature according to these dimensions, the taxonomy provides a unified framework for analyzing and comparing agent-based autonomic networking solutions.

One of the fundamental dimensions of the taxonomy is the role of agents within the autonomic networking framework. In many studies, agents are assigned specialized responsibilities to support the autonomic control loop. Monitoring-oriented agents are commonly employed to observe network conditions such as traffic patterns, resource utilization, latency, and fault indicators, providing the situational awareness required for adaptive control [3], [5]. Decision-oriented agents analyze this information and determine appropriate responses, including configuration changes, routing adjustments, or resource reallocation, often guided by predefined policies or learned strategies [4], [9].

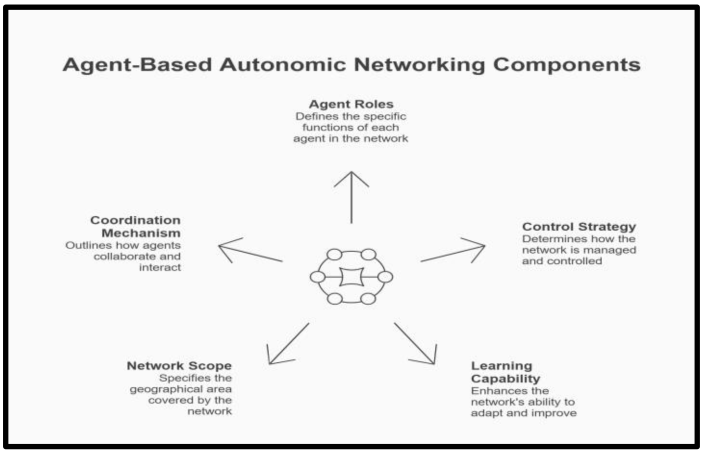

Figure 3. Conceptual Taxonomy of Agent-based Autonomic Networking Components

Execution-oriented agents are responsible for enforcing these decisions on network elements, ensuring that planned actions are applied consistently across the system. In more complex environments, coordination or negotiation agents are introduced to manage interactions among multiple agents, resolve conflicts, and align local decisions with global network objectives [5], [11]. It is important to note that these agent roles represent functional abstractions rather than strictly isolated software entities. In practical implementations, a single agent may simultaneously perform monitoring, decision-making, and execution functions, particularly in resource-constrained or tightly integrated environments. The separation of roles in the proposed taxonomy is conceptual and analytical, intended to clarify system design responsibilities and architectural patterns. This functional decomposition facilitates structured analysis and comparison across existing studies, even though real-world systems often combine multiple roles within unified agent components. Therefore, overlap among roles is not only possible but common in operational autonomic networking systems. To operationalize the taxonomy and demonstrate how representative studies align with the proposed classification dimensions, Table 2 maps selected works to key taxonomy categories.

Control strategy represents another important taxonomy dimension. Agent-based autonomic networking systems may adopt centralized, decentralized, or hierarchical control strategies depending on how decision-making authority is distributed. Centralized approaches rely on a limited number of agents with a global network view, whereas decentralized approaches distribute intelligence across multiple agents operating with local knowledge.

Table 2. Mapping of Representative Studies to Taxonomy Dimensions

Reference	Control Strategy	Learning Capability	Agent Role Focus	Network Domain
[4], [5]	Decentralized	Hybrid	Monitoring and Decision	Network Management
[9], [11]	Decentralized	Reinforcement Learning-Based	Decision and Coordination	SDN
[10]	Hierarchical	Rule-Based	Execution	Cloud
[2], [15]	Decentralized	Reinforcement Learning-Based (MARL)	Decision and Coordination	Autonomous Systems
[3], [12]	Centralized	Rule-Based	Monitoring and Execution	General Autonomic

Centralized approaches rely on a limited number of agents with a global network view, whereas decentralized approaches distribute intelligence across multiple agents operating with local knowledge. Hierarchical models combine both strategies by organizing agents into multiple coordination layers. These strategies influence scalability, coordination overhead, and system responsiveness [12], [13]. Decentralized approaches enhance scalability, robustness, and responsiveness, making them particularly suitable for large-scale and highly dynamic networks [9], [11]. Hierarchical control models represent a compromise between these extremes by organizing agents into multiple layers, where higher-level agents manage global objectives and lower-level agents handle local adaptations, thereby balancing coordination efficiency and scalability [5], [12].

The learning capability dimension classifies agent-based autonomic networking systems according to how agents acquire and update decision-making knowledge. Based on the reviewed literature, four primary categories can be identified:

1. **Rule-Based (Non-Learning) Agents:** Agents operate using predefined condition–action rules and static policies without adaptive updates during runtime. These

systems offer predictability and low computational overhead but limited adaptability to unseen scenarios [3], [13].

2. **Supervised Learning-Based Agents:** Agents employ machine learning models trained on labelled datasets, such as traffic traces or fault logs, to perform classification, prediction, or anomaly detection tasks. While effective for pattern recognition, their adaptability depends on the availability and representativeness of training data [11].
3. **Reinforcement Learning (RL)-Based Agents:** Agents learn optimal policies through interaction with the environment by maximizing cumulative reward. These approaches support dynamic adaptation in non-stationary network conditions but may face convergence and stability challenges.
4. **Hybrid Learning Agents:** Hybrid systems combine rule-based control for baseline stability with learning components (supervised or RL) for adaptive optimization. Such systems aim to balance predictability and adaptability, and are increasingly adopted in autonomic networking frameworks [4], [5].

This classification clarifies the spectrum of learning capabilities ranging from static rule-driven behaviour to fully adaptive reinforcement learning-based decision-making.

Agent-based autonomic networking solutions can also be classified according to their network scope, which determines the spatial and administrative extent over which agents operate. In localized-scope approaches, agents manage specific network segments, domains, or service clusters, enabling fast and context-aware adaptations with minimal coordination overhead [10], [14]. However, localized control may lead to suboptimal global behaviour if coordination among agents is insufficient. Global-scope techniques offer network-wide optimization and policy consistency by extending agent control across various domains or administrative boundaries, but at the expense of higher system complexity and communication overhead [12].

4. Architectures and Models

Agent-based autonomic networking systems rely on well-defined architectural designs and computational models to realize self-managing behaviour in distributed and dynamic environments. Existing studies propose a variety of architectures that differ in terms of how agents are organized, how decision-making authority is distributed, and how coordination is achieved among network entities. This section reviews the principal architectures and models adopted in agent-based autonomic networking and analyzes their suitability for large-scale and heterogeneous network environments.

4.1 MAPE-K–Based Agent Architectures

Many agent-based autonomic networking solutions are built upon the Monitor–Analyze–Plan–Execute over Knowledge (MAPE-K) framework, which provides a structured control loop for autonomic behaviour [3], [12]. In such architectures, monitoring agents continuously collect network state information, including traffic metrics, resource utilization, and fault indicators. Analysis agents process this information to detect anomalies, performance degradation, or policy violations. Planning agents use learned techniques or predefined rules to decide appropriate corrective or optimization actions based on this analysis, while execution agents implement these decisions by modifying operational parameters or reconfiguring network parts. The control loop is supported by a shared knowledge base that keeps track of management guidelines, system models, and historical data. MAPE-K–based architectures offer modularity and conceptual clarity; however, their effectiveness in large-scale networks depends on efficient inter-agent coordination and scalable knowledge management mechanisms [21].

4.2 Centralized Agent Architectures

In centralized agent architectures, a single agent or a small group of agents maintains a global view of the network and performs most autonomic decision-making functions [12], [13]. These architectures simplify policy enforcement and enable global optimization by aggregating network state information at a central point. As a result, centralized approaches often achieve consistent decision-making and reduced coordination complexity among agents. However, the reliance on a global controller introduces scalability limitations, increases response latency in large or geographically distributed networks, and creates potential single points of failure. Consequently, centralized agent architectures are generally unsuitable for highly dynamic or large-scale networking environments.

An important limitation of centralized agent architectures is their failure recovery time. When the central controller or coordinating agent fails, the entire decision-making process may be temporarily disrupted until fault detection, controller restart, and state synchronization are completed. Recovery time in such architectures typically depends on failure detection latency, controller reinitialization time, restoration of the global network state, and redistribution of pending control tasks. In geographically distributed networks, additional propagation delays may further increase recovery time.

4.3 Decentralized and Distributed Agent Architectures

In decentralized agent architecture, intelligence is distributed across multiple agents, each managing a subset of network components or operating within a specific domain [9], [11]. The agents make decisions based on the limited information available to each agent, which helps in adapting quickly to changing network environments. The distributed control approach makes it highly scalable, fault-tolerant, and responsive, making it an ideal candidate for cloud, IoT, and edge networking. However, the lack of global information leads to issues such as coordination consistency, decision conflicts, and convergence to global performance objectives. Hence, it is very important to have good communication and coordination between agents for stable and coherent system operation.

4.4 Hierarchical Agent Architectures

Hierarchical agent architectures aim to balance the advantages of centralized and decentralized control by organizing agents into multiple layers [5], [12]. In this model, higher-level agents are responsible for global policy management, coordination, and long-term optimization, while lower-level agents manage local resources and perform fine-grained adaptations. This layered structure enables scalability and local autonomy while preserving a degree of global oversight. Nevertheless, hierarchical architectures must carefully manage inter-layer communication to avoid bottlenecks and delayed responses, particularly in environments with rapidly changing network conditions.

4.5 Agent Communication and Interaction Models

The communication and interaction of these agents are of primary importance for the operation of autonomic networking systems. Agents use message-based communication to share information with other agents. They use negotiation, coordination, and consensus protocols to address any potential conflicts. Cooperative models allow agents to share knowledge and optimize network performance, while competitive models enable agents to achieve individual goals, including using conflict resolution techniques to ensure stability [11]. Although these models are beneficial for adaptability and robustness, they also cause communication overhead, especially when the network density is high.

4.6 Computational Models for Agent-Based Autonomic Networking

The type of computational model used by the agent plays a significant role in terms of adaptability and complexity in autonomic networking systems. The most commonly used models are rule-based systems, in which policies are applied, and learning-based systems, in which machine learning or reinforcement learning is utilized. Hybrid systems that combine rule-based systems and learning-based systems are also being studied. More information on this topic is provided in Section V [3], [11], [15]. Apart from convergence, another important issue in autonomic networking systems is stability and safety. Reinforcement learning is often used in autonomic networking systems, but it can sometimes result in oscillatory behavior in the agent, potentially leading to a decrease in system performance. During the training phase, if exploration is conducted by the agent, there may be a violation of service level agreements.

The application of hybrid models in computing includes the integration of specific rules with learning mechanisms, which helps in maintain a baseline level of stability while incorporating learning capabilities. The application of such hybrid models is increasing in agent-based autonomic networking systems, as it helps balance robustness and flexibility within autonomic systems, as mentioned in [4], [5]. In order to compare the architectural models described in the aforementioned paragraphs, Table II shows the relative characteristics of the described autonomic agent architectures according to specific performance criteria. The comparison is qualitative and depends on the relative architectural properties and evaluation trends mentioned in the existing literature, as specific quantitative frameworks for autonomic agent architectures have not been developed.

4.7 Agent Reasoning Models in Autonomic Networking

It is assumed that agent-based autonomic networking systems have a hybrid reasoning model that includes both reactive and deliberative reasoning models. The reactive reasoning model allows agents to respond to network events using condition-action pairs, while deliberative reasoning models, such as the belief–desire–intention (BDI) model, are used for goal-oriented planning and are consistent with the MAPE-K loop. The learning-based model uses reinforcement learning to optimize policies and choose actions that maximize long-term network performance. Most autonomic networking systems have a hybrid model of reasoning. The comparison of autonomic networking architectures, as shown in Table 3, is based on characteristics found in representative studies of agent-based autonomic networking and multi-agent network control architectures.

Table 3. Comparative Analysis of Agent-Based Autonomic Networking Architectures

Architecture	Scope	Response Latency	Scalability	Communication Overhead	Convergence Stability	Reliability & Deployment	Ref
Centralized Agent Architecture	Global	Moderate–High	Low–Moderate	Low	High	Low fault tolerance; simple deployment	[12], [13]
Decentralized / Distributed Agents	Local	Low	High	High	Moderate	High reliability; moderate deployment complexity	[9], [11]
Hierarchical Agent Architecture	Mixed (Local and Global)	Moderate	High	Moderate	High	Moderate–High reliability; high deployment complexity	[5], [12]
Pure MAPE-K Agent Systems	Depends on implementation	Moderate	Moderate	Moderate	High	Moderate reliability; moderate deployment complexity	[3], [12]
Learning-Based / MARL Systems	Local / Cooperative	Low	High	High	Low–Moderate stability	High reliability; very complex deployment	[9], [15]

From the comparative study carried out in Table III, it is evident that no particular architecture is universally optimal. Centralized architectures have strong convergence guarantees due to their global network visibility, but they are restricted in scalability and fault tolerance. Decentralized systems have minimal reaction latency and high scalability but incur high coordination costs and may experience convergence instability in dynamic environments.

Hierarchical architectures attempt to find a middle ground by incorporating hierarchical control structures, which offer high stability and scalability benefits. Learning-based and MARL-based systems provide high scalability and flexibility benefits but may have convergence instability and high computational complexity in dynamic environments. These results indicate that hybrid hierarchical learning-based techniques may have significant potential in autonomic networking.

5. Agent-Based Techniques in Autonomic Networking

Agent-based autonomic networking employs a range of techniques that enable agents to perceive network conditions, make decisions, and adapt their behavior over time. These techniques differ in terms of intelligence, adaptability, and computational complexity. This section reviews the most important agent-based techniques used to realize autonomic behavior in networking environments.

5.1 Rule-Based Agent Techniques

Rule-based agent techniques represent one of the earliest and most widely adopted approaches in autonomic networking. In this technique, agents rely on predefined condition–action rules and policies to make decisions based on observed network states. Such agents are simple to implement, predictable in behaviour, and computationally efficient, making them suitable for environments with well-defined operational conditions. However, their reliance on static rules limits adaptability, as they struggle to cope with unforeseen network dynamics or evolving traffic patterns [3], [13]. Although rule-based agent techniques provide predictable behaviour and low computational overhead, their reliance on static policies significantly limits adaptability in highly dynamic networking environments. These approaches struggle to handle unforeseen traffic patterns, evolving network conditions, or complex multi-domain interactions. Consequently, purely rule-based techniques are increasingly supplemented with adaptive learning mechanisms in modern autonomic networking architectures.

5.2 Learning-Based Agent Techniques

Learning-based agent techniques enhance autonomic behaviour by enabling agents to adapt their decision-making through experience. These agents employ machine learning or reinforcement learning methods to learn optimal actions based on feedback from the network

ISSN: 2582-1369

environment. Learning-based techniques are particularly effective in dynamic and uncertain environments, as they allow agents to adjust strategies in response to changing network conditions. Nevertheless, challenges related to training overhead, convergence time, and stability can affect their suitability for real-time network management [11], [15]. Despite their adaptability, learning-based techniques introduce several challenges for real-world deployment. Training complexity, data availability, and convergence stability can significantly affect system performance, particularly in rapidly changing network environments. Moreover, poorly designed reward functions or training datasets may lead to suboptimal or unstable decision-making behaviour, highlighting the need for robust training methodologies and safety constraints.

5.3 Multi-Agent Reinforcement Learning (MARL)

Multi-agent reinforcement learning extends single-agent learning by enabling multiple agents to learn concurrently while interacting within a shared environment. In autonomic networking, MARL allows agents to coordinate actions such as resource allocation and routing while adapting to the behaviour of other agents. This technique supports decentralised control and scalability, making it attractive for large-scale networks. However, non-stationarity introduced by multiple learning agents and increased coordination complexity remain significant challenges [9], [15]. While MARL enables distributed learning and cooperative optimization across multiple agents, it also introduces challenges related to coordination stability and scalability. As the number of agents increases, the environment becomes highly non-stationary, complicating convergence and increasing training time. Ensuring stable coordination among multiple learning agents remains an active research challenge in autonomic networking.

5.5 Hybrid Agent-Based Techniques

Hybrid agent-based techniques use a combination of methods, e.g., rule-based control and learning-based adaptation, to achieve a balance between stability and flexibility. In hybrid agent-based techniques, rule-based control ensures that agents behave in a stable and reliable manner, while learning-based adaptation helps agents cope with long-term changes in network conditions. Hybrid techniques are becoming popular in autonomic networking to overcome the limitations of individual techniques and improve overall system performance. Nevertheless, the design of effective hybrid models requires integration to avoid increased complexity and interactions between techniques that may cause conflicting effects [4, 5]. Hybrid agent-based techniques have been proposed to improve the balance between stability and flexibility in decision-making. In hybrid agent-based techniques, combining multiple decision-making techniques may result in increased architectural complexity and difficulties in maintaining consistency in decision-making.

6. Applications of Agent-Based Autonomic Networking

Agent-based autonomic networking has been applied across diverse networking domains where scalability, decentralization, and adaptive control are critical. Recent studies on knowledge-driven autonomous networks and self-healing network architectures further highlight the importance of intelligent agents for maintaining resilient network infrastructures [22]. Unlike conventional management systems, agent-based approaches embed distributed intelligence within network elements, enabling local decision-making while coordinating toward global objectives. This section critically analyzes major application domains, highlighting architectural patterns, agent techniques, evaluation practices, and existing limitations.

6.1 Network Management and Fault Recovery

One of the first and most developed application domains for agent-based autonomic networking is network management. In this context, agents have been used to facilitate decentralized MAPE-K loops, where monitoring agents are used for collecting performance metrics such as packet loss, latency, and throughput, analysis agents are used for performing anomaly detection or threshold-based evaluation, planning agents determine corrective actions, and execution agents adjust network parameters. However, recent solutions have utilized agents for fault recovery using both rule-based and learning-based agents. In the context of fault recovery, rule-based systems have been used for their predictability and low computational overhead. However, most solutions have been validated using simulation environments and testbeds, as opposed to production networks. Furthermore, coordination among multiple agents for network recovery is difficult, especially in the presence of cascaded failures.

6.2 Software-Defined Networking (SDN)

In Software-Defined Networking environments, agent-based autonomic networking is primarily employed to mitigate scalability and latency limitations of centralized SDN controllers. Traditional SDN architectures rely on logically centralized controllers, which can become bottlenecks under high traffic loads or in geographically distributed deployments. Agent-based extensions introduce distributed agents at controller clusters, edge switches, or domain-level control layers to perform local decision-making. Learning-based agents are increasingly adopted in SDN for adaptive routing, congestion control, and traffic engineering. Multi-agent reinforcement learning (MARL) frameworks allow distributed agents to coordinate flow optimization strategies while adapting to traffic fluctuations. Hierarchical agent architectures are also common, where local agents handle immediate control tasks while higher-level agents maintain global policy enforcement. Despite demonstrated scalability improvements, these approaches introduce coordination overhead and potential inconsistencies in global policy enforcement. Maintaining network-wide state coherence among distributed agents remains a significant technical challenge.

6.3 Cloud and Edge Networking

Cloud and edge environments present highly dynamic resource allocation and orchestration challenges due to workload variability, virtualization, and heterogeneous infrastructure. Agent-based autonomic networking is applied to enable decentralized resource scheduling, service placement, and load balancing across distributed nodes. In these environments, agents often operate at multiple layers: infrastructure-level agents monitor resource utilization (CPU, memory, bandwidth), service-level agents manage application deployment and scaling, and coordination agents enforce service-level objectives. Learning-based models, particularly reinforcement learning and predictive analytics, are commonly employed to anticipate workload changes and proactively adjust resource allocation. Decentralized and hierarchical agent architectures are widely adopted in cloud–edge ecosystems to balance scalability and coordination efficiency. While such architectures improve responsiveness and reduce orchestration latency, they introduce complexity in inter-agent communication and consistency management. Moreover, integration with existing orchestration frameworks (e.g., Kubernetes-based environments) remains non-trivial, and interoperability across heterogeneous cloud providers is still an open research issue. Evaluation practices in this domain often focus on performance metrics such as resource utilization efficiency and response time, but rarely address long-term stability, convergence guarantees, or security implications of autonomous agent behaviour.

6.4 Internet of Things (IoT) Networks

IoT networks consist of large numbers of resource-constrained and geographically distributed devices, making centralized management inefficient and often impractical. Agent-based autonomic networking enables IoT gateways or edge nodes to act as autonomous agents, performing local adaptation such as dynamic routing, energy-aware scheduling, and context-aware communication. Most IoT-focused agent systems adopt decentralized architectures due to device heterogeneity and limited bandwidth. Lightweight rule-based agents are commonly used for energy-efficient decision-making, while learning-based agents are applied in more capable edge nodes for adaptive optimization. However, IoT deployments introduce significant constraints related to energy consumption, computational capacity, and security vulnerabilities. Agent communication overhead must be carefully managed to avoid excessive energy drain. Furthermore, trust management and secure coordination among agents are critical concerns, as IoT nodes are often exposed to adversarial environments. Despite promising simulation-based results, large-scale real-world validation remains limited.

6.5 Vehicular and Autonomous Networks

Vehicular and autonomous network environments demand ultra-low latency, high mobility support, and real-time decision-making. Agent-based autonomic networking is used to facilitate coordination in distributed environments. Agents use multi-agent reinforcement learning to facilitate routing, spectrum allocation, and driving strategies. Agents can modify communication parameters in response to topology changes and density changes. A decentralized architecture is best suited to this environment, as a centralized architecture will not meet the ultra-low latency requirements. Vehicular networks face issues of extreme non-stationarity due to their high mobility, which makes convergence difficult in learning-based environments. Stability, safety, and predictability in vehicular environments are still research challenges. Additionally, verification of agent-based decision-making in real-time environments is still an emerging research area.

6.5.1 Cross-Domain Observations

Across all application domains, several patterns emerge:

- Decentralized and hierarchical architectures dominate due to scalability requirements.
- Hybrid agent models combining rule-based and learning-based techniques are increasingly adopted.
- Most evaluations rely on simulation or small-scale testbeds.
- Coordination overhead and convergence guarantees remain insufficiently addressed.
- Security, trust, and interoperability are recurring open issues.

The transition from research prototypes to large-scale production networks requires advances in verification frameworks, standardized communication protocols, scalable coordination mechanisms, and realistic experimental platforms. Overall, the reviewed application domains demonstrate that agent-based autonomic networking can improve adaptability and resilience across diverse network environments. However, most existing implementations remain limited to experimental or simulation-based evaluations. Bridging the gap between research prototypes and real-world deployment requires improved verification methods, standardized coordination protocols, and scalable evaluation platforms.

7. Challenges and Open Research Issues

In spite of the promising features that agent-based autonomic networking offers, some challenges that inhibit the full adoption of agent-based autonomic networking, especially in large-scale autonomous network environments. These challenges arise from the complexities associated with autonomous decision-making, coordination, and trust issues. Some of the major challenges affecting the full adoption of agent-based autonomic networking include scalability. In agent-based autonomic networking, scalability refers to the efficiency of agent-based systems in large-scale environments without compromising their performance. The number of agents used in the system increases significantly in proportion to the size of the network. Therefore, scalability remains a major research challenge in agent-based autonomic networking, especially in large-scale network environments. Another major challenge is coordination and convergence. In decentralized and multi-agent systems, agents tend to make decisions based on local and partial information, which can cause conflicts in the decision-making.

8. Conclusion and Future Research Directions

This survey on agent-based computing for autonomic networks has examined various agent-based computing paradigms for autonomic networks by studying their architectures, techniques, and domains using a taxonomy. From this survey, several limitations have been identified. However, various unresolved issues have also been identified in this survey. These unresolved issues include scalability in multi-agent systems, convergence, security and trust in autonomous systems, predictability, interoperability with existing infrastructures, and finally, a lack of standardized evaluation environments. In this regard, future work on agent-based computing for autonomic networks must focus on strong security and trust environments for autonomous systems, scalable and lightweight agent systems, and finally, on hybrid agent systems that balance learnability and stability. In order to improve predictability and facilitate evaluation in real-world scenarios, formal verification techniques must also be developed. This is important for agent-based autonomic networks to move from experimental to dependable systems, thereby paving the way for fully autonomous, resilient, and intelligent systems in next-generation digital infrastructures.

References

[1] Arzo, Sisay Tadesse, Riccardo Bassoli, Fabrizio Granelli, and Frank HP Fitzek. "Multi-Agent Based Autonomic Network Management Architecture." IEEE Transactions on Network and Service Management 18, no. 3 (2021): 3595-3618.

[2] Jing, Peng, Hanbin Hu, Fengping Zhan, Yuexia Chen, and Yuji Shi. "Agent-Based Simulation of Autonomous Vehicles: A Systematic Literature Review." IEEE Access 8 (2020): 79089-79103.

[3] Choi, Jin Seek, and Se Joon Chun. "AgentVisor: Agent-Based Network Hypervisor for Autonomic Network Virtualization." IEEE Network 37, no. 6 (2022): 158-165.

[4] Arzo, Sisay Tadesse. "Towards Network Automation: A Multi-Agent Based Intelligent Networking System." (2021)

[5] Sifakis, Joseph, Dongming Li, Hairong Huang, Yong Zhang, Wenshuan Dang, River Huang, and Yijun Yu. "A Reference Architecture for Autonomous Networks: An Agent-Based Approach." arXiv preprint arXiv:2503.12871 (2025)

[6] Chekati, Adil, Meriem Riahi, and Faouzi Moussa. "Agent-based Modelling Approach for Decision Making in an IoT Framework." In International Conference on Advanced Information Networking and Applications, Cham: Springer International Publishing, 2021, 261-272.

[7] Pati, Ashis Kumar. "Agentic AI: a comprehensive survey of technologies, applications, and societal implications." IEEE Access (2025).

[8] Acharya, Deepak Bhaskar, Karthigeyan Kuppan, and B. Divya. "Agentic AI: Autonomous Intelligence for Complex Goals—A Comprehensive Survey." IEEe Access 13 (2025): 18912-18936.

[9] Serugunda, Claire Naiga, Sisay Tadesse Arzo, Fabrizio Granelli, Riccardo Bassoli, Michael Devetsikiotis, and Frank HP Fitzek. "Autonomous Network Traffic Classifier Agent for Autonomic Network Management System." In 2021 IEEE Global Communications Conference (GLOBECOM), IEEE, 2021, 1-6.

[10] Alti, Adel, and Lamri Laouamer. "Agent-Based Autonomic Semantic Context-Aware Platform for Smart Health Monitoring and Disease Detection." The Computer Journal 65, no. 3 (2022): 736-755.

[11] Liu, Peini, Gusseppe Bravo-Rocca, Jordi Guitart, Ajay Dholakia, David Ellison, and Miroslav Hodak. "Scanflow-k8s: Agent-Based Framework for Autonomic Management and Supervision of Ml Workflows in Kubernetes Clusters." In 2022 22nd IEEE International Symposium on Cluster, Cloud and Internet Computing (CCGrid), IEEE, 2022, 376-385.

[12] Dennis, Louise A., and Michael Fisher. "Verifiable Self-Aware Agent-Based Autonomous Systems." Proceedings of the IEEE 108, no. 7 (2020): 1011-1026.

[13] Dimopoulos, Filippos, Edgar Eler, Marieke Timmers, Jan Treur, and Sander L. Koole. "An Agent-Based Network Model for Interpersonal Emotion Regulation in a Human-Bot Interaction." In International conference on computational collective intelligence, Cham: Springer Nature Switzerland, 2023, 68-79.

[14] Geng, Sunyue, Sifeng Liu, and Zhigeng Fang. "An Agent-Based Algorithm for Dynamic Routing in Service Networks." European Journal of Operational Research 303, no. 2 (2022): 719-734.

[15] Yadav, Pamul, Ashutosh Mishra, and Shiho Kim. "A Comprehensive Survey on Multi-Agent Reinforcement Learning for Connected and Automated Vehicles." Sensors 23, no. 10 (2023): 4710.

[16] https://www.slideshare.net/slideshow/autonomic-computing-152721696/152721696.

[17] Savaglio, Claudio, G. Fortino, R. Gravina, and W. Russo. "A Methodology for the Development of Autonomic and Cognitive Internet of Things Ecosystems." In Proc. IEEE Int. Conf. Cloud Eng., 2017, 317-322.

[18] Liu, Teng, Xiaohong Huang, and Kun Xie. "Towards Autonomous Network Management: An Intelligent and Secure Agentic Framework for Network Configuration." IEEE Network (2025).

[19] Zhang, Zhiyang, Fangkai Yang, Xiaoting Qin, Jue Zhang, Qingwei Lin, Gong Cheng, Dongmei Zhang, Saravan Rajmohan, and Qi Zhang. "The vision of autonomic computing: Can llms make it a reality?." arXiv preprint arXiv:2407.14402 (2024).

[20] Wu, Binghan, Shoufeng Wang, Yunxin Liu, Ya-Qin Zhang, Joseph Sifakis, and Ye Ouyang. "Leveraging AI Agents for Autonomous Networks: A Reference Architecture and Empirical Studies." arXiv preprint arXiv:2509.08312 (2025).

[21] Cegielski, Wendy H., 'Networks, Agent-Based Modeling, and Archaeology', in Tom Brughmans, and others (eds), Network Simulation, in Tom Brughmans, and others (eds), The Oxford Handbook of Archaeological Network Research, Oxford Handbooks (2023; online edn, Oxford Academic, 20 Nov. 2023), https://doi.org/10.1093/oxfordhb/9780198854265.013.16, accessed 16 Mar. 2026.

[22] Fang, Honglin, Peng Yu, Can Tan, Junye Zhang, Dahua Lin, Liyan Zhang, Yong Zhang, Wenjing Li, and Luoming Meng. "Self-Healing in Knowledge-Driven Autonomous Networks: Context, Challenges, And Future Directions." IEEE Network 38, no. 6 (2024): 425-432.

[23] Kandel, Asmita, Ihsan Ibrahim, and Naoki Fukuta. "An Analysis of Educational Cloud Platforms Using Multi-Agent Learning." In 2022 12th International Congress on Advanced Applied Informatics (IIAI-AAI), IEEE, 2022, 230-233.