# Autonomic Computing and VANET

James J. Mulcahy
Computer & Electrical Engineering
and Computer Science
Florida Atlantic University
Boca Raton, FL, United States
jmulcah1@fau.edu

Shihong Huang
Computer & Electrical Engineering
and Computer Science
Florida Atlantic University
Boca Raton, FL, United States
shihong@fau.edu

Imad Mahgoub
Computer & Electrical Engineering
and Computer Science
Florida Atlantic University
Boca Raton, FL, United States
imad@cse.fau.edu

***Abstract*— As modern wireless communication networks continue to spread in coverage and ubiquity, so do the applications for networks that take advantage of mobile technology. One of the more interesting areas of research and development is in the development and deployment of vehicular ad hoc networks (VANETs). VANETs offer the potential for intelligent transportation networks that can both actively and passively improve travel efficiency and safety for the vehicles that use them. Informative content can be delivered to drivers informing them of road conditions or nearby traffic congestion. Entertaining content like multimedia can be delivered to vehicle passengers. To be usable and efficient, VANETs need to be largely autonomous and self-adaptive. The software that organizes the nodes entering and leaving a VANET must be self-managing, without requiring their active participation of drivers or passengers in the organization and maintenance of the network. A VANET needs to automatically adapt to changes in the geography over which the network is deployed, and to the highly dynamic behavior of its participant vehicles. It must detect and recover from failures during content delivery, protect the integrity of data being delivered, and protect against malicious attacks by intruders. It should optimize the exchange of information between participant vehicles and roadside infrastructure, making the best use of power and bandwidth. In this paper, we describe the architecture of vehicular ad hoc networks, the principles of autonomic software design, and how the autonomic computing paradigm is applied to VANET applications.**

***Keywords—wireless sensor networks, vehicular ad hoc networks, software engineering, autonomic computing, self-adaptive systems, self-managing systems, MAPE-K, control loops***

## I. INTRODUCTION

Specialized wireless sensor networks like intelligent transportation systems (ITS) and vehicular ad hoc networks (VANETs) are enjoying an increase in popularity in both application and research. There are more than one billion passenger cars on the road world-wide today [1]. This number is projected to swell to four billion by 2050 [2]. The improvement and ubiquity of increasingly cheaper hardware allows modern vehicles to become rolling computing and storage devices. With increasing sophistication and coverage of wireless communication networks, especially those that are composed of mobile nodes like vehicles, it is no wonder that VANETs suffer from a problem of scale and complexity.

Government entities seek to improve public safety and make traffic management more efficient and reliable. Commercial entities are interested in improving the experience and comfort of the vehicle occupants, supplying navigation details, time-sensitive information and news, and multimedia entertainment for passengers ("infotainment") [3]. Researchers and practitioners strive to improve and evolve techniques used to deploy and maintain ad hoc networks, especially to improve their flexibility, efficiency, reliability, and security. To achieve these goals, VANETs exhibit a significant degree of autonomy to provide a seamless experience for human drivers and passengers.

"Autonomic computing" is a term coined by IBM in 2001 to describe a design approach that imparts self-managing behavior to software systems, aimed at reducing or mitigating system complexity, while increasing reliability. The approach involves the development of systems capable of managing their own resources and behavior with little or no human intervention, given certain high-level system objectives defined by humans [4]. Since modern hardware and software can be inextricably linked, this term naturally extends to the usefulness of autonomic systems that are a synthesis of both hardware and software. Wireless sensor networks (WSNs), mobile ad hoc networks (MANETs), and vehicular ad hoc networks (VANETs) are examples of types of systems that stand to benefit from autonomic design. Ultimately, the most efficient VANETs are those that operate completely autonomously, responding to changes in the environment in which they operate in a self-configuring, self-optimizing, self-protecting, and self-healing manner.

While there is a body of literature that addresses the use of the autonomic computing in the development of new software systems and the extension of legacy software systems, and literature that describes wireless sensor networks like MANETs and VANETs, this paper aims to provide a synthesis that describes how VANETs exhibit autonomic design and behavior as described by IBM. The remainder of the paper is organized as follows: Section II provides background about wireless sensor networks and autonomic computing. Section III describes vehicular ad hoc networks in the context of autonomic computing, while Section IV concludes the paper and offers ideas for future work in this area of engineering.

## II. BACKGROUND

Nearly one hundred and forty years have passed since Alexander Graham Bell uttered the first words over a telephone, summoning Thomas Watson from his laboratory [5]. Today the world is connected by wired and wireless infrastructure. High-capacity fiber optic cables are draped along the bottom of the world's oceans. Fiber, copper, and coaxial cables are buried into our infrastructure and strung between our buildings. The development of satellite-to-ground communications and mobile packet-based networks in the 1960s and 1970s began to untether us from wired infrastructure [6]. More recently, radio, cellular, Wi-Fi, Bluetooth, and a host of other wireless technologies have appeared. Mobile sensors and devices have become cheaper and more ubiquitous, and access to a variety of mobile-based service without being limited by timing or location is now possible [7].

### A. *Wireless Sensor Networks*

A wireless sensor network (WSN) is a distributed network of sensor nodes that has little or no permanent infrastructure [8]. In a WSN, the nodes themselves make up the structure of the network. The nodes can send, receive, and store data, and may act as routers of data packets destined for other distant nodes. (Fig. 1). WSNs are composed of a large number of sensors that can be deployed quasi-randomly and densely over a selected area of interest, and collectively used to monitor and exchange information about the surrounding environment [9]. Wireless sensors typically exchange data with other sensors via short range radio. Memory storage and power are limited in most applications, and consequently a modern engineering challenge is to produce sensors that consume as little bandwidth and power as possible. This drives the need for efficient distribution of data over sensor networks.

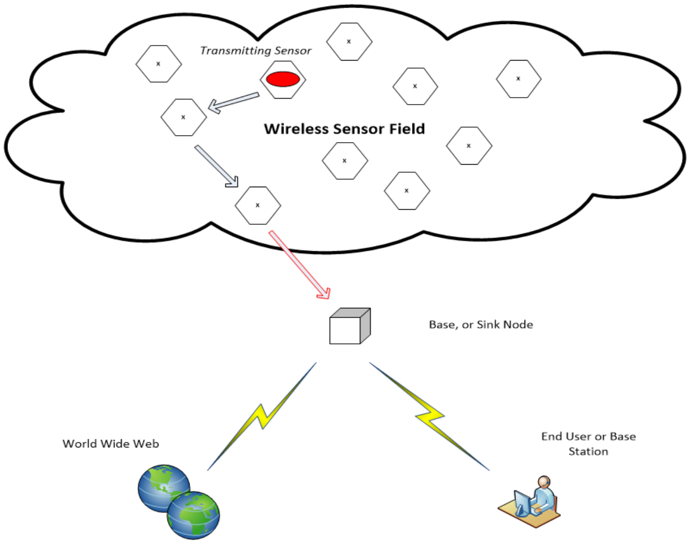

**Fig. 1. Multi-hop wireless sensor network (WSN) example.**

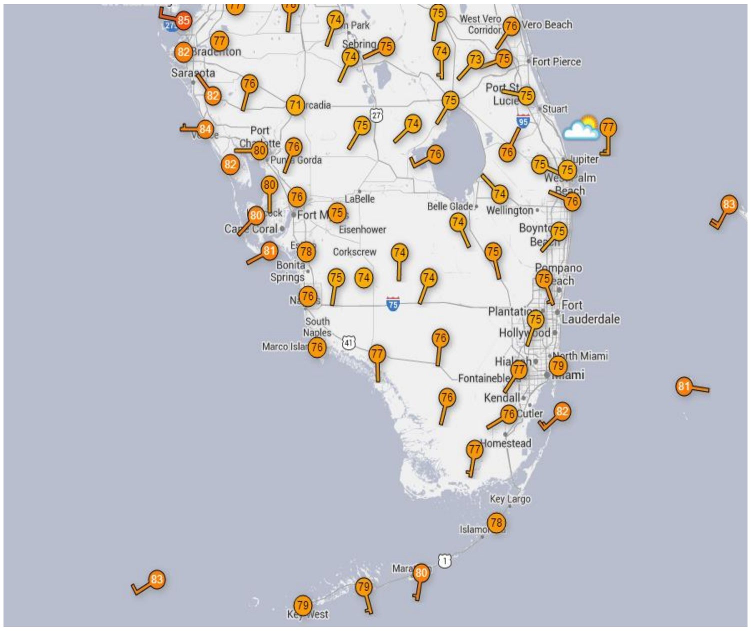

**Fig. 2. Wireless weather station network reporting in real-time.**

WSNs have been around in some form since at least the 1950s, when the United States military deployed a network of submerged acoustic sensors to detect and track Soviet submarines [10]. By the 1970s, the U.S. Army was developing packet-switching technology and packet radio systems [11]. There are many applications for statically place WSNs, like monitoring air quality, the structural integrity of buildings [12], water temperature and composition of inland waterways [13]. WSNs can be used to monitor forest fires [14], seismic conditions [15], and even weather conditions over large geographical areas (Fig. 2). Other WSNs are designed for industrial applications that monitor and collect data from power plants, water treatment plants, factories, and data storage facilities [16]. In each of these examples, the sensor nodes are typically placed once and not subsequently relocated.

### B. *MANETs and VANETs*

Mobile ad hoc networks (MANETs) and vehicular ad hoc networks (VANETs) are designed to perform similar types of sensory and communications tasks as WSNs, but composed of mobile nodes. VANETs are a special class of MANETs, composed of nodes represented by vehicles rather than people, floating buoys, or other less mobile or semi-mobile platforms [17]. MANETs and VANETs must both adapt to a wide variety of geographic topologies, and a dynamically changing set of mobile participant nodes. At any given moment, a node in a MANET or VANET can travel into or out of the range of other neighbor nodes and of any nearby fixed network infrastructure. In MANETs, the movement of the nodes is highly variable, and relatively slow. A battlefield scenario in which the participant nodes are soldiers wearing sensor and communication equipment is an example of a MANET [18].

A VANET differs from a MANET with respect to node mobility and the topology of the network. In a VANET, nodes are vehicles, and move in more predictable directional patterns. Automotive vehicles typically travel along well-defined roadways, but at a much higher rate of speed [19].

Physical data storage is less of a concern in the implementation of VANETs than other wireless sensor networks. Unlike humans carrying mobile devices that are necessarily limited in weight and size, automotive vehicles can support heavier and more robust computing devices with higher-capacity memory, computing power, communication capability, and power storage capacity. Power is less of a concern, since most vehicles are likely to contain alternators or other power-generating equipment. With a constant power source when the vehicle is running and access to a battery for backup when the vehicle is off, on board units (OBUs) are less affected by power limitations of other kinds of mobile devices [20].

Like other wireless sensor networks, VANET nodes share data wirelessly via radio transmission. Communication between participating vehicles is known as "Vehicle to Vehicle (V2V)." "Vehicle to Infrastructure (V2I)" refers to data transmitted from individual vehicles to roadside units, and "Infrastructure to Vehicle (I2V)" refers to communication that originates with an RSU but is targeted at one or more participating vehicles. The vehicle OBUs may in turn re-broadcast information to other vehicles within its range, extending the reach of the network beyond the physical range of the RSU. Data targeted for a specific recipient may be communicated directly when the sender and receiver are in range of each other, referred to as "single hop", or it may be communicated by routing it through one or more other vehicles in a "multi-hop" process (Fig. 3). For example, traffic information data that is broadcast from a fixed roadside unit (RSU) reaches all "listening" vehicle nodes within range of that RSU.

This may not be the most efficient and reliable way to route the data, however. If the target vehicle is beyond range of the RSU, for example, it may be impossible. A more efficient and reliable to send the data in several concurrent pieces, or "packets," via multiple "hops" using intermediary vehicles as routers between the source and target of the data. Efficient delivery of content is necessary for a reliable network that does not consume significant portions of the available bandwidth, particularly when the network topology changes frequently [21].

To be useful, reliable, and efficient, a VANET must be able to reconfigure itself when the topology of the network changes. Individual vehicle nodes may have varying types of OBUs with varying storage capacities, processing power, and communication capability. The highly dynamic nature of a VANET makes this organizational task too difficult for human participants to maintain.

Within a VANET, communication between nodes is a challenge of efficiency and reliability. To reduce consumption of limited transmission bandwidth, a VANET must optimally route data between nodes quickly and efficiently.

VANETs must have the ability to recover from faults without disrupting the rest of the network. A vehicle suddenly leaving the network with data that it hasn't yet been forwarded to a recipient node is an example of a fault. There are many reasonable scenarios in which a vehicle node may leave a VANET abruptly. For example, there may be a failure in transmission capability, such as a faulty antenna [22]. Another example is when a vehicle may travel beyond the range fixed infrastructure and of all other vehicles that are members of the network. In another scenario, a vehicle may crash and damage the OBU or experience hardware failure in the OBU itself. A third – and more common – scenario is described by a driver reaching his or her intended destination and subsequently shutting down the vehicle. Unless the OBU is designed to communicate with the network on battery power, the node disappears from the network when the vehicle is powered off.

A VANET must protect itself from malicious attacks originating outside the network. They may be aimed at disrupting network operation, corrupting the data transmitted between nodes, or intercepting private data meant for another vehicle. In short, a VANET must have a high degree of autonomicity, which can be accomplished with a combination of hardware architecture and software.

Node discovery, efficient data routing, recovery from communication failures, and protection from malicious attacks are but a few of the requirements of a vehicular ad hoc network [19]. In this paper, these four requirements are used to illustrate the self-*, or self-managing attributes that VANETs inherently must exhibit to have value to its human participants.

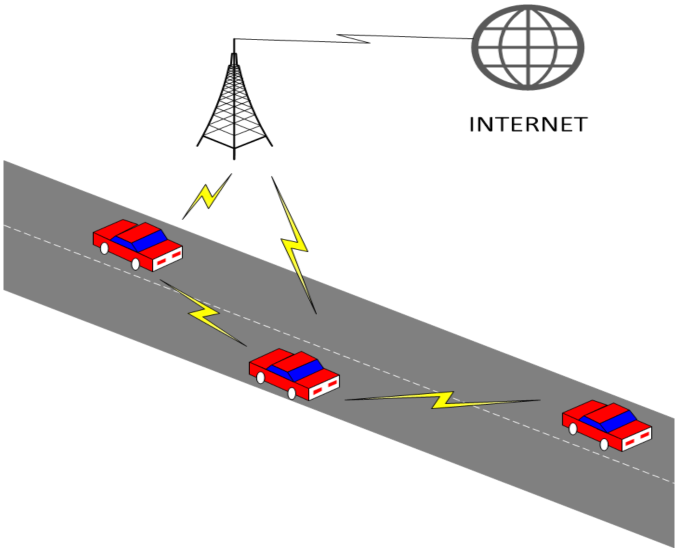

**Fig. 3. A simplified model of a multi-hop VANET.**

### *C. Autonomic Computing*

In 2001, IBM published a “manifesto” that addressed the growing complexity of software systems. The manifesto coined the term “autonomic computing” that defines the attributes of a system that reduces reliance upon humans to monitor, configure, optimize, correct, and protect software systems. Autonomic systems exhibit self-* (“self-star”) behavior, meaning they have “adaptivity” properties [23]. An autonomic system is also self-governing, or “self-managing” [24]. Self-managing systems have each of the four properties outlined in IBM’s manifesto, and are described as follows:

The *self-configuring* property refers the ability of a system to detect and adapt to changing conditions in or requirements by its environment. A self-configuring system may automatically add or remove software components of the system in real time without waiting for human intervention. For example, the operating system of a personal computer may install or remove driver software when it discovers newly attached hardware peripherals or other software, anticipating what a human would do in its stead. In this regard, the operating system is said to be self-configuring. In industry, a commercial enterprise resource planning system (ERP) may be designed to monitor external computing systems, some of them owned by other stakeholders. They need to be able to recognize when connectivity to external systems are lost or restored, and react accordingly. They may monitor local or external data stores, and reacting to the appearance of particular types of data. A case study is presented in [25] that describes an autonomic solution for a commercial retailer. Its self-configuring behavior included the detection and processing of purchase orders by web-based customer orders without any other human monitoring or action.

*Self-optimizing* systems automatically monitor and tune themselves with the goal of improving the quality of system attributes like reliability and efficiency. For example, operating systems use automated process schedulers to achieve high system utilization rates while executing applications according to priority, and this is done without a human operator needing to intervene. In commercial systems, internal and external resources are monitored and resource allocations altered according to pre-defined rules or parameters designed to keep the system operating within specific performance requirements. In sensor networks, optimal shortest-distance or shortest-time paths may be computed for optimal routing of data to and from origin and destination nodes.

A *self-healing* system is one that recognizes and recovers from system faults without any human assistance. Similar to the concept of “fault tolerance,” self-healing is a more robust behavior than the simple recovery from a fault [26]. The goal of a self-healing system is to return it back to normal operation once the fault is detected and handled [23]. Modern virus detection software is an example of application-level solutions that monitor for, detect, and quarantine malicious software to keep the host system operating normally. Commercial ERPs can be designed to exhibit self-healing behavior by detecting and adapting to unplanned lapses in connectivity with other systems they communicate with [27].

*Self-protecting* systems are designed to anticipate system faults, recover from human error, and prevent malicious attacks from external sources, acting automatically to prevent or mitigate further damage [23]. This behavior is proactive, while a self-healing behavior reactive. A self-protecting system actively monitors itself in order to detect threats to system integrity as they occur, before a system fault occurs. An anti-malware application on a personal computer, for example, may automatically “blacklist” a website that is known to infect target machines with viruses or worms, or it may quarantine a virus as soon as it is recognized and before it can damage any data.

These four self-* attributes describe the behavior of an autonomic system. At the root of the design of autonomic systems is the architectural model that helps impart these attributes. An autonomic system may consist of one or more *autonomic components* that interact with each other. A component can itself be further composed of other autonomic components. Components can be inputs to yet other autonomic components. Autonomic components consist of the resource being managed and the software that manages that resource.

An *autonomic manager* is a software mechanism that monitors software or hardware (or both), decides how to alter the system, and automatically applies the changes needed to alter that behavior. The autonomic manager is a control loop that intelligently monitors a managed element, and adapts its behavior accordingly. A more complicated or inclusive autonomic manager can be designed to impart several or all of the self-* attributes to system or subsystem in which it operates. A common model used to describe the architecture of the control loop that a typical autonomic manager employs upon a managed element is the MAPE-K model described by IBM [4].

### *D. The MAPE-K Model*

MAP-K is an acronym that represents the major components of an autonomic control loop architecture, and is illustrated in Fig. 4. The *monitoring* component allows the autonomic manager to detect – or sense – changes in its operating environment, using hardware sensors or software-based detection schemes.

The *analysis* component determines whether the collected data indicates a change in the system worthy of action. The *planning* component determines what action or actions should be taken, and the *execution* component applies necessary changes to the system or managed resource. The *knowledge* component stores information that is gathered about the element being managed, the environment in which it operates, and the rules that govern expected system behavior. Information can include historical operational data that is later mined and used to more intelligently and finely tune system behavior.

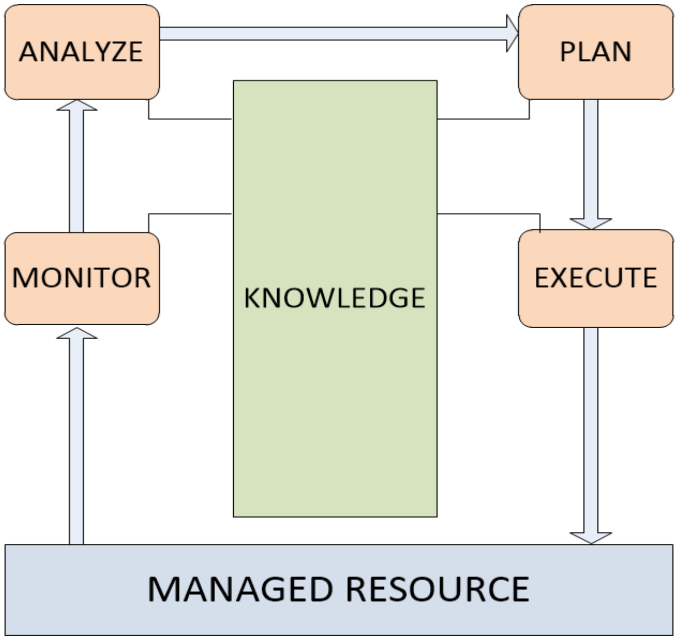

**Fig. 4. MAPE-K control loop model.**

## III. AUTONOMIC COMPUTING AND VANET

The autonomic computing paradigm described by IBM addresses the growing complexity of hardware and software systems. At the most abstract level, an autonomic manager controls a managed resource. Managed resources may include individual computers, servers, and data storage devices, but they also may include operating systems, commercial software applications, and the middleware that connects and integrates them. A managed resource may even be an entire business process, like a commercial order fulfillment systems workflow in [25]. VANETs are themselves a blend of hardware and software components and processes that inherently need to be autonomic.

In a VANET, the nodes (OBUs and RSUs) are the autonomic elements. A vehicle's OBU *monitors* its environment using sensors attached to the vehicle or its occupants, and monitors a communication channel for broadcasts from roadside units and other vehicles (I2V and V2V). A RSU *monitors* the network to detect vehicles entering and leaving its range, and listens for broadcasts from the individual OBUs (V2I).

Nodes in a VANET are tasked with *optimizing* the appropriate routing path for data transmission between infrastructure and vehicles, and between vehicles. Optimization reduces network overhead like power consumption and bandwidth usage, while improving the reliability of the network by reducing data collisions and packet losses, and reacting to interference. The network must *heal* from data loss by detecting when a participant vehicle has left the network without delivering its data payload to the next vehicle in an active routing path. The data must be rebroadcast and routing tables need to be updated to exclude the missing vehicle. OBUs and RSUs must be designed to detect and *protect* from malicious attacks upon the network, while preserving the integrity and confidentiality of the data being transferred between participating nodes.

Like other autonomic systems, VANETs must have self-managing characteristics for the network to be useful to its users, and should not require humans to actively configure, operate, or maintain the network. The self-configuring, self-organizing, self-healing, self-protective attributes of an autonomic VANET must be seamless and invisible to the humans operating vehicles in the network. The following subsections elaborate upon the autonomic attributes that VANETs have.

### A. A VANET is Self-Configuring

Vehicular ad hoc networks automatically respond to changes in their operating environment. A VANET's operating environment consists of a set of participating vehicles that form the nodes of the network, plus the set of roadside infrastructure units that communicate with them. RSUs are generally permanent infrastructure and are therefore geographically static nodes. Vehicles, however, may enter and leave the network at any time with no forewarning. They may change direction, change speed or acceleration, and may even be shut down or powered on while in range.

A self-organizing, or *self-configuring* VANET detects changes in the network topology and makes the appropriate adjustments, like modifying routing tables to reflect new participant vehicles or departing vehicles. A vehicle will typically leave a network by traveling outside the range of the RSUs, and outside the range of the closest vehicle that still maintains contact with the network.

Alternatively, a vehicle may reach its intended destination while still in range of the network, and disappear from the network when the driver shuts it down. Conversely, vehicles that are powered up within range of a VANET or otherwise enter the network must be accounted for. In worst-case scenarios, a vehicle can crash and damage its OBU, or the OBU equipment could fail for other reasons. Connectivity between the vehicle and the network can also fail, effectively removing the vehicle from being seen by other nodes in the network.

### B. A VANET is Self-Optimizing

A self-configuring VANET automatically handles the scaling of the network from just a few vehicles (or even one) to dozens or hundreds of vehicles or more. A self-optimizing network transmits data between participating vehicles and infrastructure as efficiently as possible, even as the topology constantly changes. A self-optimizing network is able compute the most efficient routing of the data the goal of reducing network overhead and increasing the reliability of the exchange of data between nodes.

Topology-based and geographical-based are two types of routing schemes used to optimize the flow of data in a

VANET. Topology-based proactive protocols like Fisheye State Routing are table-driven. Reactive protocols like Ad Hoc on Demand Distance Vector (AODV), Dynamic Source Routing (DSR), and Temporally Ordered Routing Protocol (TORA). Geographical-base routing protocols include Delay Tolerant Network (DTN), Greedy Perimeter Stateless Routing (GPSR), and Vehicle-Assisted Data Delivery (VADD). The advantages and disadvantages of each protocol is beyond the scope of this work, but are presented to show how VANETs are designed to be self-optimizing [20].

### C. *A VANET is Self-Healing*

Vehicles typically leave a VANET with no warning. If a departing vehicle is in possession of one or more data packets that are destined for another vehicle, those packets are lost. A *self-healing* VANET recognizes that a packet has been lost, updates its routing tables to exclude the now-missing node, and retransmits the failed packets. In table-based routing models, the tables are updated accordingly. In this manner, the network has "healed" itself by first detecting the fault and then taking action to restore the intended function of the network – delivering of data to target nodes.

### D. *A VANET is Self-Protecting*

While self-healing describes a reactive behavior, self-protecting describes both reactive and proactive behaviors. A *self-protecting* system is able to anticipate the possibility of a fault or failure before it occurs, and is able to automatically take steps to prevent or block the problem from occurring or causing more damage. For example, a VANET may be designed to compute the speed and orientation of each vehicle in its range, rerouting data packets through interior nodes rather than edge nodes to their intended destination, to increase the changes that the data arrives at its intended destination without interruption. While VANETs protect themselves from data loss, they must also protect themselves from intentional attacks from sources external to the network. Some attacks are intended to disrupt or corrupt the communication of data across the network, or to overwhelm the network's resources. The latter is known as a "denial of service" (DoS) attack [29].

Other malicious techniques include "Sybil" attacks in which a malicious node broadcasts information to multiple neighboring vehicles, spoofing each with a fake identity [30]. Node impersonation is another example in which a malicious node pretends to be another vehicle [31]. A malicious node may broadcast false information to impersonate another node, or to have an effect on traffic flow in a biased or otherwise disruptive way. A self-protecting VANET is one that is able to successfully detect the presence of a malicious node or vehicle and block it from participating in the network or minimizing its impact.

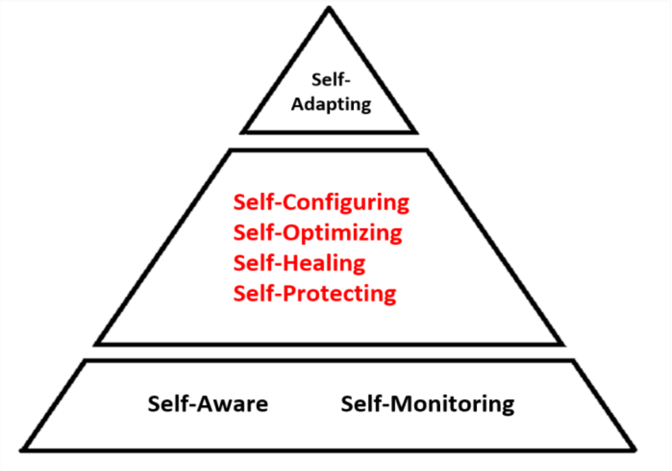

**Fig. 5. Three-level self-* hierarchy.**

The four attributes of a VANET discussed in this section collectively describe an autonomic system as defined by IBM. Other researchers have elaborated on the autonomic architecture, expanding it beyond the four basic attributes identified by IBM (Fig. 5). In [23], Salehi and Tahvildari describe a three-level hierarchy of self-* properties. The top level is the *self-adaptive* or *self-organizing* attribute, a generalized term to describe all of, or the result of, the self-* behaviors of an autonomic system. The middle level consists of the four major properties from IBM: *self-configuring, self-optimizing, self-healing,* and *self-protecting.* The third and most primitive level of attributes are those that describe the behavior of a system that is *self-aware* and *context-aware*. In VANET this is accomplished by self-monitoring and limiting communication between trusted participating vehicles and roadside infrastructure over a discrete geographical range.

## IV. SUMMARY & FUTURE WORK

In this paper, we described the autonomic computing paradigm, detailing four major attributes that define the behavior of such systems. We described the general structure of wireless and vehicular ad hoc networks, and how VANETs by design manifest the attributes of an autonomic system. The examination of the autonomic structure of VANETs in this paper is limited, and future work will include a more thorough review and evaluation of the existing and planned applications of VANET and autonomic design.

Future work will also explore how the autonomic design approach in VANETs may be extended to other modes of commercial travel. Perhaps a day approaches when airlines, rail traffic, boat traffic, and vehicular traffic will all be interlinked into one seamless, mode-agnostic autonomous transportation network.

## V. References

[1] "Automobiles & Trucks Business Trends Analysis," *Plunkett Research Ltd.*, 11 November 2014. Web. 01 March 2015. <http://www.plunkettresearch.com/trends-analysis/automobiles-hybrid-electric-business-market/>.

[2] L. Chou, J. Tseng, and J. Yang, "Adaptive Virtual Traffic Light Based on VANETs for Mitigating Congestion in Smart City," *The Third International Conference on Digital Information and Communication Technology and its Applications (DICTAP2013)*, The Society of Digital Information and Wireless Communication, 2013.

[3] P. Salvo, F. Cuomo, A. Baiocchi, and A. Bragagnini, "Road side unit coverage extension for data dissemination in VANETs," in *Wireless On-Demand Network Systems and Services (WONS), Proc. of the 9th Annu. Conf. on,* pp. 47-50, January 2012.

[4] IBM, "Architectural blueprint for autonomic computing," http://www-03.ibm.com/autonomic/pdfs/AC%20Blueprint%20White%20Paper%20V7.pdf 2005. Web. 1 November 2014.

[5] T. Watson, "How Bell invented the telephone," *American Institute of Electrical Engineers, Proceedings of the,* vol. 34, no. 8, pp. 1503-1513, August 2015.

[6] A. Butrica, Ed., *Beyond the Ionosphere: Fifty Years of Satellite Communication,* NASA Special Publication-4217, 1997.

[7] K. Chung, J. Yoo, and K. Kim, "Recent trends on mobile computing and future networks," *Personal and Ubiquitous Computing*, vol. 18, no. 3, pp. 489-491, 2014.

[8] J. Garcia-Macias and J. Gomez, "MANET versus WSN," in *Sensor Networks and Configuration: Fundamentals, Standards, Platforms, and Applications,* pp. 369–388, 2006.

[9] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393-422, March 2002.

[10] R. Holler, "The Evolution of the Sonobuoy from World War II to the Cold War," No. JUA-2014-025-N, NAVMAR Applied Sciences Corp, 2014.

[11] M. Gerla and L. Kleinrock, "Vehicular networks and the future of the mobile internet*," Computer Networks*, vol. 55, no. 2, pp. 457-469, February 2011.

[12] T. Fu, A. Ghosh, E. Johnson, and B. Krishnamachari, "Energy-efficient deployment strategies in structural health monitoring using wireless sensor networks," *Structural Control and Health Monitoring*, vol. 20, no. 6, pp. 971-986, 2013.

[13] Zennaro, Marco, et al. "On the design of a water quality wireless sensor network (WQWSN): An application to water quality monitoring in Malawi," *Parallel Processing Workshops, (ICPPW'09), Int. Conf. on*, pp. 330-336, September 2009.

[14] Y. Aslan, I. Korpeoglu, and Ö. Ulusoy, "A framework for use of wireless sensor networks in forest fire detection and monitoring," *Computers, Environment and Urban Systems*, vol. 36, no.6, pp. 614-625, November 2012.

[15] Liu, Guojin, et al., "Volcanic earthquake timing using wireless sensor networks," *Information Processing in Sensor Networks , Proc. of the 12th Int. Conf. of.,* pp. 91-102, 2013.

[16] S. Huang and X. Zhao, "Application of wireless sensor networks on power plants monitoring," *Applied Mechanics and Materials*, vols. 321-324, pp. 762-766, June 2013.

[17] R. Raut, P. Thakare, and R. Bhoyar, "Conspectus of Various Routing Protocols in VANET," *International Journal of Advent Research in Computer & Electronics*, vol. 1, no. 2, 2014.

[18] G. Thomeczek, I. Colwill, and E. Stipidis, "Mission aware topology healing for battlefield MANET," *Journal of Battlefield Technology*. vol. 17, no. 3, December 2014.

[19] M. Sood, S. Kanwar, "Clustering in MANET and VANET: A survey," *Circuits, Systems, Communication and Information Technology Applications (CSCITA), 2014 International Conference on*, pp. 375-380, April 2014.

[20] B. Paul, M. Ibrahim, and M. Bikas, "VANET Routing Protocols: Pros and Cons," *International Journal of Computer Applications,* vol. 20, no. 3, pp. 28-34, April 2011.

[21] S. Zeadally, R, Hunt, Y. Chen, A. Irwin, and A. Hassan, "Vehicular ad hoc networks (VANETS): status, results, and challenges," *Telecommunication Systems*, vol. 50, no. 4, pp. 217-241. December 2010.

[22] S. Worrall, G. Agamennoni, J. Ward, E. Nebot, "Fault Detection for Vehicular Ad Hoc Wireless Networks," *Intelligent Transportation Systems Magazine, IEEE*, vol.6, no.2, pp.34-44, April 2014.

[23] M. Salehie and L. Tahvildari, "Self-adaptive software: Landscape and research challenges," *ACM Transactions on Autonomous and Adaptive Systems*, vol. 4, no. 2, pp. 14:1-14:42, May 2009.

[24] J. Kephart and D. Chess, "The vision of autonomic computing," *Computer* , vol. 36, no. 1, pp. 41-50, January 2003.

[25] J. Mulcahy and S. Huang, "Autonomic Software Systems: Developing for Self-Managing Legacy Systems," *Software Maintenance and Evolution (ICSME)*, *2014 Int. Conf. on*, pp. 549-552, October 2014.

[26] D. Ghosh, R. Sharman, H. Raghav Rao, and S. Upadhyaya, "Self-healing systems — survey and synthesis," *Decision Support Systems*, vol. 42, no. 4, pp. 2164-2185, January 2007.

[27] J. Mulcahy, S. Huang, and A. Veghte, "Leveraging service-oriented architecture to extend a legacy commerce system," *Systems Conference, 2010 4th Annual IEEE*, pp.243-248, April 2010.

[28] M. Huebscher and J. McCann, "A survey of autonomic computing: degrees, models, and applications," *ACM Computing Surveys,* vol. 40, no. 3, pp. 1–28, August 2008.

[29] R. Raw, M. Kumar, and N. Singh, "Security challenges, issues and their solutions for VANET," *Int. Journal of Network Security & Its Applications (IJSNA)*, vol. 5, no. 5, September 2013.

[30] B. Yu, C. Xu, B. Xiao, "Detecting Sybil attacks in VANETs," *Journal of Parallel and Distributed Computing,* vol. 73, no. 6, pp. 746-756, June 2013.

[31] G. Guette, and C. Bryce, "Using tpms to secure vehicular ad-hoc networks (VANETs)," *Information Security Theory and Practices. Smart Devices, Convergence and Next Generation Networks,* vol. 5016, pp. 106-116, 2008.