La Résistance: Harnessing Heterogeneous Resources for Adaptive Resiliency in 6G Networks

Ganesh C. Sankaran  
*Information Sciences Institute*  
Los Angeles, CA, USA  
gsankara@isi.edu

Joaquin Chung  
*Argonne National Laboratory*  
Lemont, IL, USA  
chungmiranda@anl.gov

Raj Kettimuthu  
*Argonne National Laboratory*  
Lemont, IL, USA  
kettimuth@anl.gov

Abstract—Recent years have seen more critical applications designed to protect human lives (e.g., environmental sensing, emergency response, and tactical defense) being deployed over wireless networks. These critical deployments expect higher data rates, ultra-low latency, and ultra-high reliability. 6G wireless networks are expected to fill the gap in terms of the first two aspects (i.e., higher data rates and ultra-low latency), however providing ultra-high reliability is a wide open challenge. All is well when everything works but when there is a failure or a security attack, the entire system collapses exposing the associated human lives to imminent danger. Avoiding this requires the strongest of assurances that safety and security aspects are protected no matter what. Large scale critical applications are waiting for this piece of the puzzle to be solved. At this juncture, we envision the bold theme of La Résistance 6G (LR6G) that would pave the way for deploying mission-critical applications and services over 6G networks. It aims to achieve ultra-high reliability and resiliency to any disruptions, be it failures or security attacks. A few disruptions are easy to handle such as a cloud VM or primary link failure. In both of these cases, applications can be restored by activating the standby resource. However, some disruptions can be detrimental such as when a cut-vertex fails or when the disruption leaves the critical application to fail without access to a standby resource. These critical applications (e.g., Smart Manufacturing, Smart City, Ocean monitoring, Wildfire monitoring, etc.) are highly distributed in nature. They must continue to deliver their mission objectives during a disruption to protect human lives. In this paper, we present our LR6G vision and outline the challenges towards achieving this vision.

Index Terms—6G networks; adaptation; heterogeneous resources; resiliency

I. INTRODUCTION

The sixth generation (6G) of wireless mobile networks are expected to have key performance indicators such as 1 Gbps data rates (per user), ultra-low latency (1 ms or less), massive numbers of devices, and ultra-high reliability (99.99999%). 6G will also offer significant improvement in programmability and quality of service (QoS) by leveraging technologies such as software-defined networking (SDN), network function virtualization (NFV), multi-access edge computing (MEC), in-network computing, dynamic orchestration, and machine-learning/artificial-intelligence (ML/AI). All these features make 6G networks a complex infrastructure that provides a plenty of opportunity to incorporate non-traditional resiliency capabilities in the network for networked applications.

Systems built to protect human lives are increasingly deployed over communication networks. Examples of these systems are the smart city, smart agriculture, power grid, autonomous vehicle traffic systems, tactical defense networks, and emergency response systems. When communication breaks or is delayed, it puts human lives in danger. For example, delay in notifying a frequency drift in a smart grid network will not start the power generation at the right time and cause the entire grid to collapse in a ripple effect.

The current generation of networks (are expected to) meet their mission objectives under normal operating conditions, but we envision 6G-enabled systems (with an architecture that exploits the advanced features intelligently such as the one proposed here) will be capable of meeting mission objectives even during disruptions. In order to handle a wide variety of adversarial and failure conditions, resiliency should be a pillar of 6G system design and architecture. In the context of 6G systems, resiliency can be defined as the ability to survive, gracefully adapt to, and rapidly recover from malicious attacks, component failures, and natural and human-induced disruptions.

To provide resiliency, the traditional approach is to replicate a component so that when one instance goes down, the rest can handle new requests [2]. However, replication is limited to homogeneous stack environments containing similar hardware, operating system, libraries, etc. This work takes a step beyond the current state-of-the-art to restore the critical functionality. We propose to restore critical functionality by adapting it on to the available non-homogeneous resources in the device-to-edge-to-cloud continuum of 6G networks.

We envision a La Résistance 6G (LR6G) network with adaptive resiliency. To motivate this, let us consider the following example. A critical function/service hosted on the C-RAN is disrupted due to a security (say reflective DDoS) attack. The Edge-to-Cloud network is flooded and is dropping all benign packets as suspicious. The only resilient option is to harness a few small IoT devices to put up a valiant fight against the adversary.

5G network relaxes the association of network function to a part of the network (e.g., 5G Core or NG-RAN). 5G operator chooses a placement that optimizes their objective function. The 5G network does not yet understand the notional loss resulting out of the service disruption. 5G network handles...
the following disruptions in this example scenario:

a) Connectivity disruption case: The 5G network or the edge network attempts to establish alternate connectivity. This is part of path diversity and multi-homing approaches.

b) Cloud resource disruption case: Cloud manager migrates the function to a new cloud resource. This is more a disruption-centric best effort approach. When notional loss is high, restoring the service becomes super-critical. For instance, service disruption in a nuclear power plant would cause a radiation exposure threat in the whole region e.g., Fukushima disaster. With 6G, IoT devices are expected to take up more and more mission-critical that tasks directly impact human lives.

LR6G network is aware of its criticality and adapts itself to restore services in non-traditional ways. First of all, LR6G network expands its scope beyond network functions and is more system- and service-centric. LR6G has the following distinctive characteristics:

1) It creates a unified resource pool of the available resources be it network equipment, storage equipment, or IoT devices without loosing visibility into their individual capacity, strengths and weaknesses.

2) It creates a unified programmable platform out of heterogeneous resources to meet desired performance and security objectives.

3) It has the ability to slice and dice the functionality to fit individual resource capacity constraints. While doing so results in an increase in the attack surface. The unified platform will have the ability to create suitable checks and balances to identify future issues arising due to any weak links across the system.

The rest of the paper is organized as follows: Section II provides background on resiliency strategies for distributed functions. Section III presents our proposed La Résistance 6G (LR6G) architecture. Finally, Section IV presents our conclusions.

II. BACKGROUND

Self-healing networks heal themselves without a human intervention [3]. As and when disruption occurs, the network detects the situation and makes the necessary changes in an automated manner to heal itself. Drawing a cue from self-healing networks, our LR6G system will restore its functionality without human intervention.

Today, networked applications (such as smart city, smart agriculture, autonomous cars) are built to meet specific societal objectives with components distributed across the network [4]–[6]. A set of distributed functions that is essential to meeting these objectives are henceforth known as critical functions. These functions are not only restricted to the application layer but also include security, safety, and critical network layer functions. During a network failure, a portion of the network nodes lose communication from the rest. Such failure may be the effect of a variety of causes including but not limited to (i) equipment misconfigurations [7], (ii) component aging [8], or (iii) a systematic attack [9]. This is likely to happen more often in the far edge of wireless networks. As 6G is expected to operate in higher bands of the spectrum, signals are more directional, more susceptible to interference, requiring more intelligent management of the spectrum.

Currently, a given component (function) is mapped to its stack (underlying hardware, operating system, and software libraries) at design time and any major change is a time-consuming effort [10]. In the device-to-edge-to-cloud continuum of 6G, it may be difficult to find homogeneous resources. We envision to restore functionality using non-homogeneous stack. However, this is a challenging task as this new pool of resources may be resources constrained or may have non-traditional architecture and programming paradigms. The following subsections provide details on our approach.

A. Gradual Service Degradation

Fig. 2 represents gradual service degradation, a paradigm that networked applications could leverage. When there is
cloud connectivity, sufficient resources are available to support full functionality. When connectivity to cloud fails, reachabil-
ity up to Fog is present. In this situation, less than sufficient resources are available. With these resources the network architecture should lose a part of the functionality but should not cripple in terms of operation. Similarly, when fog or edge connectivity is lost, the functionality and resources are further reduced. With the available resources, the system shrinks to keep as much functionality alive. This includes the critical functionality. Based on resource availability, more high priority functionality is supported.

Fig. 2. Gradual service degradation as a function of resource availability and functionality

To take advantage of gradual degradation during a disruption, a 6G-enabled system must monitor its components for failures and security compromises, compute a sequence of functions that will meet the mission objectives, and adapt these functions to the available resources. However, key challenges must be addressed for deploying resilient 6G-enabled systems:

- State-of-the-art monitoring mechanisms can be either complex or simple, with complex mechanisms requiring considerable amount of resources and simple mechanisms lacking in accuracy. A light-weight highly accurate monitoring mechanism is required to detect performance and security anomalies in the system on a continuous basis.
- Current resiliency mechanism rely on homogeneous full-stack resources for resiliency by replication. However, this limits the number of available resources that are able to maintain critical functions operations. The Cloud has abundant homogeneous resources, but the rest of the device-to-edge-to-cloud continuum lacks abundant homogeneous resources.
- By making heterogeneous resources available, we could increase the pool of resources that critical functions could use to remain operative. However, converting a cloud-function to a edge-function or a in-network function is not straightforward. Furthermore, when one-to-one function-to-resource mapping is not found, the function must be collectively realized by many resources.
- The problem of choosing where and how to redeploy a critical function for resiliency can rapidly increase with the number of requirements. These requirements may include meeting function resource demands with available resources; resolving any critical, security and safety function sequencing conflicts; and meeting the performance and security objectives.

In general, a programmable data plane pipeline may perform a function on the data stream that would have otherwise been executed on an end node. It computes a result (e.g., future hardware mapping and placement of a function ) that will be consumed by an application (e.g., computer vision or augmented reality). In previous work [11] we demonstrated how to leverage in-network computing for performing scientific operations (that are not supported in network devices) using approximations in a streaming fashion.

As we will discuss in Section III, programmable data plane pipelines could be used to host critical functions in response to a disruptive event (e.g., attack or failure). However, many challenges remain open for this technology to be a real enabler of resiliency in 6G networks and systems.

III. ENVISIONED LR6G SYSTEM

We propose La Résistance 6G (LR6G), a dynamic architecture for 6G systems that adapts 6G networked applications to meet critical security and safety requirements during a disruption. We envision

1) a self-restoring system that restores and retains its critical functionality in the event of a disruption,
2) a unified pool of resources that can be mobilized in the event of a disruption,
3) a software-defined distributed system that provides a unified programming model to program a wide variety of devices, and
4) the ability to survive disasters (or) in other words survival to $N - \epsilon$ failure

Being adaptive is a control loop where the current situation is sensed at first and then the system reacts to the current situation to meet the mission objectives. We partition our adaptive vision into four broad areas viz. (1) ASSESS, (2) MONITOR, (3) COMPUTE, and (4) ADAPT. Here, the first two map to static and dynamic analysis of the situation, while the last two map the system’s reaction towards service restoration.

At a high level, the individual areas will encompass the following:

(i) ASSESS: assess available hardware resources (network switches, accelerators, and IoT devices) on a range of security and performance parameters. This static analysis will help in determining the capacity of the available unified resource pool.
(ii) MONITOR: monitor resource utilization to detect security and performance anomalies in a granular fashion.

(iii) COMPUTE: compute the optimal function sequence based on the current network state (failure and attack state), available resource capacity, the resource constraints (in terms of performance and security) and semantic relationships. This will compute the mapping of code fragments onto the available resource pool.

(iv) ADAPT: We envision a unified programming model to distribute and program the critical functions onto the available resources.

We now present how each of these areas and associated challenges.

A. ASSESS

We envision ASSESS to provide greater visibility into the available resource pool creating a unified view. This unified view must encompass various dimensions of the resources such as power, performance and security. Understanding these dimensions will enable us create a minimally vulnerable or minimally susceptible system.

Device layers such as hardware, operating environment, and applications may have different vulnerabilities or be susceptible to different types of failures. The key challenge is to have an accurate view of these attributes for every device. This is combinatorial and it is difficult to assess every device or every (hardware, OS, application) combination. An orthogonal view of hardware, OS, and application is useful to represent a device without an individual assessment but such a view will not be able to capture cross-layer vulnerabilities. For instance, studying the hardware (say ARM core) in isolation will be able to cover a variety of devices using this hardware model. Similarly, study of OS (say tinyOS) will cover a wide range of devices running this OS. This study is insufficient to cover the vulnerabilities with the ARM-tinyOS combination. With the volume and heterogeneity of IoT devices, the scale of this challenge balloons up.

Fig. 4 presents an example in which three system implementations (shown in different colors) are considered. The system on the left is built on top of hardware $h_a$, operating system $o_a$, and in programming language $l_a$. Each of these layers is known to have some vulnerabilities. A higher number of red triangular skull icons in the layer means that more vulnerabilities are known. For instance, hardware $h_a$ is known to have more vulnerabilities than hardware $h_c$. Thus, hardware $h_c$ is considered more secure. Both performance and security must be studied to see how hosting specific functions or a part of it can improve or degrade the critical application.

New vulnerabilities are reported every day. ASSESS should keep pace with new vulnerabilities and in the best case should be able to identify deployment specific vulnerabilities. Furthermore, the wireless channel of 6G may be susceptible to new type of vulnerabilities in the physical layer.

Another key challenge is isolation. The chosen hardware that is adapted will host critical functions in addition to the existing functions it is already hosting. Any problem encountered by the new function must not impact existing functions it is hosting and vice-versa. Isolation helps in containing any further damage to the available resources. Most devices other than cloud computing devices do not support isolation or virtualization. Moreover IoT devices are resource-constrained and it is difficult to realize the desired level of isolation over
As already mentioned, our vision is to restore the critical functions. This involves mobilizing IoT devices and/or other network resources such as RAN that are otherwise not meant to run a critical function. A few vendors lock the programmability for security reasons. The programming model of these devices are very different.

Once we detect a vulnerability in a device or a system, how quickly can we plug it is a key challenge. In this context, event-based programming models used in DataOps pipelines are very agile. The entire pipeline can be modified and upgraded on the go. A similar capability is envisioned.

**B. MONITOR**

This area keeps the system grounded to the current network state. Without sufficient monitoring, a disruption of critical functionality on a 6G system may go undetected. Response time largely depends on the time taken by monitoring considering rest of the times to be a constant. The aim is to detect the system disruption in real-time or much better if it is predicted. This is a huge challenge and a responsibility on monitoring area.

Detailed monitoring is achieved at the increase in cost of adding intrusive instructions to the devices. Many anti-virus software slow down compute hardware by introducing more monitoring instructions per executed instruction on an average. As we consider resource-constrained IoT devices, monitoring must be timely but at the same time resource friendly. This is a challenge.

Accuracy of detection must be high. Otherwise, monitoring loses its credibility. False positives are seen as a source of DoS attack on the resources. The same level of accuracy must be maintained even when prediction mechanisms are used. False negatives both impact the system detection ability and will leave the system in a broken state.

When the system is adapted to restore critical functions, new monitoring hooks will be required. For instance, a critical application when hosted in a cloud environment can afford sophisticated monitoring ability. However, when the same functionality is mapped to a resource-constrained device such as an IoT device, the same monitoring hooks will not just fit. It is challenging to reintroduce equivalent resource friendly monitoring without manual intervention.

Let us consider a situation where monitoring is mapped onto device ‘X’ that is along the network path. However, when the system is adapted, where functionality is hosted on a different device, this device ‘X’ may not be on the network path. Thus a realignment is required as and when the functionality mapping is revisited. All monitoring functions and their mapped resources must be well-known. It must be budgeted for whenever the functional mapping is computed. Realignment is challenging without manual intervention.

**Detect system level anomalies:** As noted already, it is challenging to define mechanisms that detect anomalies at a granular level without incurring high cost. Monitoring response time to detect anomalies [12], [13] is at one end of the spectrum. Producing a digest capturing every instruction executed along with CPU cycles is at the other end of the spectrum. Response time is a coarse measure that enables identification of huge deviations and the digest is the most granular that can help identify even minute deviations. However, the cost of instruction level digest is high. It is a challenge to choose the right mechanism with the right balance.

**Detect disruptions within a distributed function sequence:** When the resource demands do not fit a single node, the function is distributed across nodes (. When realizing a function across nodes, monitoring must be strengthened to identify failures and potential disruptions at the earliest and to restore the system. These disruptions may include security vulnerabilities or the 6G wireless channel. Designing mechanism to obtain information on forwarding decision for packet flows is a promising approach to build a quick isolation mechanism that is capable of identifying the faulty component in a sequence of distributed functions [14]. Furthermore, keeping track of the wireless network status and mapping it to the services impacted would be the key for our vision.

**C. COMPUTE**

This area takes all available input factors to compute the mapping of critical functionality to available resources. This computation is trivial when the given resource pool is homogeneous and individual resource capacity is at least the maximum resources required to host any critical functionality.

Here are the key challenges that must be considered during this computation.

1) As we discussed already, a LR6G resources exhibit more heterogeneity. This obviously indicates that their resource capacities are not equal. The individual resource capacity and resource requirements of the critical functionality have no relation. In many cases, functional requirements
are more than capacity. This problem is no longer trivial and must consider fragmentation.

2) The side-effect of fragmentation is distribution. Attack surface increases when the functionality is distributed. Attack surface increases when a function is distributed. The successor device should be able to identify whether the information received is really from its predecessor device and it is valid input. Predecessor device can get corrupted or spoofed. So, additional monitoring hooks to detect performance and security compromises such as execution time, proof of benign computation to detect any malicious code traversal, and proof of benign data in terms of validation of its input are required.

3) Heterogeneity increases the dimensions of the problem manifold. Volumetric attributes such as capacity constraint make this problem look more like knapsack. Categorical attributes such as resistance to security attacks makes it more like a Max-SAT problem. The state in terms of the security attacks that the system is facing is critical. One manifestation of the problem is to associate individual SAT clauses with different weights and maximize the satisfied weights.

4) Critical safety and security functions are mapped to fractional variables that add up to one when a function is fully implemented. The corresponding constraints must be satisfied outside the SAT clauses for security. Performance metrics must be satisfied as well. Metrics such as latency are non-linear in a multi-commodity flow formulation.

5) Multi-hop wireless topology must be considered when distributing functions across a set of devices from a proximity point of view. Renewable resources do not impact system lifetime. Similarly, on-tap resources if any available must be factored into the computation.

A formulation considering the above challenges will provide insights into the time complexity of the problem. We conjecture that this computation problem is NP-Hard. Response time includes computation and computing a feasible solution over a longer duration in the order of minutes can worsen the situation. Just solving the security part of the problem involving a large number of categorical variables will take much more time. Suitable heuristics must be defined to find a feasible solution quickly.

D. ADAPT

This area starts where COMPUTE ends. It articulates challenges involved in taking the functional mapping result from COMPUTE to create the binaries and configuration all the way up to complete the deployment.

1) The new function that is added to the devices must not impact their existing functionality in terms of performance or security or availability. Many resource-constrained devices are used and it is not feasible to build a strong isolated environment such as a sandbox or a container or a virtual machine that does not impact the rest of the device.

2) Heterogeneity has been mentioned as a challenge in other areas but here the lack of uniform programming and deployment model creates a host of issues. Creating a binary for a device without human intervention becomes a huge challenge. The code written for a non-real-time OS does not go well with a real-time OS target. We lack a common compiler toolchain that can create binaries for all device targets. For many small devices, OS and application are built together into a monolithic binary. Even standard library functions (e.g., standard I/O) are not available across all targets. Some platform compiler toolchains are not available publicly. In a few cases, the programming model is different for the hardware platform. Specifically, hardware that are based on GPU, FPGA, and NPU based devices. Usually portable libraries are used to be able to port the code to similar target devices but porting them across all targets found in a 6G environment is an open challenge.

3) Configuration interfaces are not uniform. The configuration interface for an IoT device is not similar to the BBU configuration interface. Especially given the volume and wide variety of IoT devices, lack of common configuration interfaces is still an unsolved challenge.

IV. CONCLUSION

We motivated the need for critical systems that impact human lives more than any natural disaster such as life-critical systems used in hospitals. These systems must survive no matter what. We then presented our vision of La Résistance 6G (LR6G) system viz. the vision for self restoration of critical services, ultra-high reliability, unified view of available resources, uniform programming model and deployment model. Finally, we presented challenges in realizing this vision. Some challenges are wide open and involve multiple stakeholders. The entire 6G ecosystem must come together to solve these challenges.

ACKNOWLEDGMENTS

REFERENCES

[1] W. Saad, M. Bennis, and M. Chen, “A vision of 6g wireless systems: Applications, trends, technologies, and open research problems,” IEEE Network, vol. 34, no. 3, pp. 134–142, 2020.

[2] W.-Z. Zhang, I. A. Elgendy, M. Hammad, A. M. Iliyasu, X. Du, M. Guizani, and A. A. A. El-Latif, “Secure and optimized load balancing for multilayer iot and edge-cloud computing systems,” IEEE Internet of Things Journal, vol. 8, no. 10, pp. 8119–8132, 2021.

[3] M. Wang, G. Simon, L. A. Neto, A. Bella, I. Amigo, L. Nuaymi, and P. Chancelou, “Sdn-oriented disaggregated optical access node for converged 5g mobile and residential services,” in 2021 European Conference on Optical Communication (ECOC). IEEE, 2021, pp. 1–4.

[4] Y. Li, L. Ma, Z. Zhong, F. Liu, M. A. Chapman, D. Cao, and J. Li, “Deep learning for lidar point clouds in autonomous driving: A review,” IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 8, pp. 3412–3432, 2021.

[5] A. Kirintat, O. Krejcar, A. Kertesz, and M. F. Tasgetiren, “Future trends and current state of smart city concepts: A survey,” IEEE Access, vol. 8, pp. 86448–86467, 2020.

[6] O. Friha, M. A. Ferrag, L. Shu, L. A. Maglaras, and X. Wang, “Internet of things for the future of smart agriculture: A comprehensive survey of emerging technologies,” IEEE CAA J. Autom. Sinica, vol. 8, no. 4, pp. 718–752, 2021.
[7] S. K. R. Kakarla, A. Tang, R. Beckett, K. Jayaraman, T. Millstein, Y. Tamir, and G. Varghese, “Finding network misconfigurations by automatic template inference,” in 17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20). Santa Clara, CA: USENIX Association, Feb. 2020, pp. 999–1013. [Online]. Available: https://www.usenix.org/conference/nsdi20/presentation/kakarla

[8] A. M. M. Paing, “Analysis of availability model based on software aging in SDN controllers with rejuvenation,” in 2020 IEEE Conference on Computer Applications (ICCA). IEEE, 2020, pp. 1–7.

[9] K. Bhardwaj, J. Chung, and A. Gavrilovska, “Towards IoT-DDoS prevention using edge computing,” in USENIX Workshop on Hot Topics in Edge Computing (HotEdge 18). Boston, MA: USENIX Association, Jul. 2018. [Online]. Available: https://www.usenix.org/conference/hotedge18/presentation/bhardwaj

[10] F. Chirigati, “Porting software without affecting scientific results,” Nature Computational Science, vol. 1, no. 3, pp. 172–172, 2021.

[11] G. C. Sankaran, J. Chung, and R. Kettimuthu, “Leveraging in-network computing and programmable switches for streaming analysis of scientific data,” in The 7th IEEE International Conference on Network Softwarization (IEEE NetSoft 2021), June 2021, pp. 1–5.

[12] Q. Kang, A. Agrawal, A. Choudhary, A. Sim, K. Wu, R. Kettimuthu, P. H. Beckman, Z. Liu, and W.-k. Liao, “Spatiotemporal real-time anomaly detection for supercomputing systems,” in 2019 IEEE International Conference on Big Data (Big Data), 2019, pp. 4381–4389.

[13] A. Sivanathan, D. Sherratt, H. H. Gharakheili, A. Radford, C. Wijenayake, A. Vishwanath, and V. Sivaraman, “Characterizing and classifying IoT traffic in smart cities and campuses,” in 2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2017, pp. 559–564.

[14] G. C. Sankaran, “Obtaining information on forwarding decisions for a packet flow,” Oct. 19 2010, uS Patent 7,817,636.