doi:10.20944/preprints201808.0237.v1

Article

Cognitive Packet Networks for the Secure Internet of Things

Joanna Domańska [1,†,‡], **Tadeusz Czachórski** [1,‡], **Mateusz Nowak** [1,‡], **Slawomir Nowak** [1,‡]

[1] IITIS-PAN, Bałtycka 5, 44-100 Gliwice, Poland; {joanna,tadek,mateusz,snowak}@iitis.pl
* Correspondence: joanna@iitis.pl; Tel.: +48 32 231 73 19
† Current address: IITIS-PAN, Bałtycka 5, 44-100 Gliwice, Poland
‡ These authors contributed equally to this work.

---

1. Introduction

Viewed for a long time as a peripheral issue, Cybersecurity is now at the forefront of everyday computer system and network operations, and of research in Computer Science and Engineering. Indeed cyberattacks, even when they are detected and mitigated, have a very large cost to systems operations including the degradation of the commercial image or trust of the and in 2017 the European Union published its recommendation for security and privacy. In addition, the organisations that operate systems that come under cyberattack can not only lose market share and lose the trust of the end users, but they also have to increase their operating costs both in terms of means to defend themselves but also in increased energy consumption and operating costs [1] and CO^2 impact [2,3].

The SerIoT Project [4] finds its origins in early work started over a decade ago on Distributed Denial of Service (DDoS) Attacks [5,6] and on using routing with the Cognitive Packet Network protocol (CPN) [7] to detect DDoS, and trace the attacking traffic so as to use CPN's ACK packets to drop the attacking traffic packets at upstream routers that carry the attacking traffic, and also detect worm attacks [8–11]. More recently, the EU FP7 Project NEMESYS [12–14] provided the opportunity to examine the cybersecurity of mobile networks including the control plane which is used to establish and keep track of calls. Since the control plane is a critical element that enables the mobile network to function, some attacks aim in particular at this part of the system [15,16]. Further work on the security of cyber-physical systems has considered vulnerabilities that address the physical infrastructure, the decision algorithms that manage the system when it operates normally or under threat, and the communication system used to convey data, information and commands between different system components [17–22].

1.1. European Research in Cybersecurity in Recent Years

In [23], some recent research on cybersecurity in Europe has been summarised regarding the several projects funded by the European Commission. A core issue that diffuses through all layers of information technology concerns cybersecurity for mobile telephony. Because most modern mobile phones offer opportunistic access [24] to WIFI and other wireless networks, the resulting security vulnerabilities should be constantly monitored both at the network and control plane levels, and in the mobile device. Thus recent [25] has investigated the use of neural network and machine learning methods to discuss this issue. The research in [26,27], addresses attacks that manipulate the signalling plane of the backbone network and directly concerns the mobile network operator as well as the end user, and the project NEMESYS addressed many of these issues [28,29] using techniques from Queuing Theory [30,31].

KONFIDO [32] concentrates on the security of communications and data transfers for interconnected European national or regional health services. Since travellers from European countries must often access health services in another European country, the health informatics systems will have to access remote patient data in a secure manner [32] and the related technical and ethical issues are addressed in a series of recent papers [33–35].

The GHOST project [36] addresses security in the IoT system market [37] for homes, and focuses on the design of a secure home IoT gateway including the attack detection techniques [38], and the analysis of attack methods that try to bring down the energy supply of the devices by draining their batteries [39]. The detection techniques that are proposed are based on Deep Learning [40] and recurrent Random Neural Neworks [41,42] that have been used previously in a variety of applications [43,44]. GHOST also investigates blockchain based methods to track and improve the security of the home IoT system[45].

2. The SerIoT Project

The SerIoT project started in January of 2018 [4], and further details regarding can also be found in a forthcoming paper [46]. The project's Technical Objectives include means to understand the threats to a IoT based economy and understand how distributed ledgers (Blockchain) may improve IoT based systems. It will design and implement virtualised self-aware honeypots to attract and analyse attacks.

The project will design SerCPN [47], a network that manages specific distributed IoT devices based on the Cognitive Packet Network (CPN). It will use the implementation of Software Defined Networks (SDN) based on CPN [48–50] using measurements that create the system self-awareness [7,51–54]. These SDNs will use "Smart" Packets (SP) to search [21,55] for secure multi-hop routes having good quality of service (QoS) and measure their security and performance, and will use Reinforcement Learning with Random Neural Networks [56] to improve the network overall performance, including all three criteria of high security, good QoS and low energy consumption [57,58]. It may be possible to extend these schemes with genetic algorithms which use an analogy between network paths and genotypes [59–61]. Several SerCPN network clusters may be interconnected via end overlay network [62], with adaptive connections to Cloud and Fog servers [63,64] for network data analysis and visualisation.

Combining energy aware routing and QoS [65,66] with security, we can also address network admission [67] to enhance security. Wireless IoT device traffic may also be specifically monitored and adaptively routed in a similar manner as it accesses SerCPN [68–70].

The project will deliver a number of platforms that comprise the main technical outputs of the project, including Platforms for (i) IoT Data Acquisition, (ii) Ad-hoc Anomaly Detection, (iii) Interactive Visual Analytics and Decision Support Tools, and (iv) Mitigation and Counteraction that will orchestrate, synchronise and implement the decisions taken by the various components.

3. Use Cases

The SerIoT project's outcomes will be evaluated in a number of significant use cases. These include four main areas. The first one is Surveillance, where physical security in bus depots will be monitored through the infrastructure of OASA which is the largest transport authority in Greece. The second one involves Intelligent Transport Systems in Smart Cities, in particular in areas such as collision avoidance, where we will demonstrate how SerIoT can enhance the cybersecurity of such systems with infrastructures proveded by OASA, Austria-Tech (ATECH), and TECNALIA for vehicle safety. The third use case will involve Flexible Manufacturing Systems (Industry 4.0), which will monitor physical attacks to wireless sensor networks in Industry 4.0 with the help of DT/T-Sys., for situations related to automated warehouses where different attack vectors may be used for breaking or jamming communication lines. The fourth use case will address Food Chains which require end-to-end security through multiple communication channels, including device authentication, detection and avoidance of DDoS and replication attacks, and detection of functionality anomalies and disabling of IoT devices. In the food chain, IoT devices may be critical to notify perishability of food items that use visually readable labels by IoT devices to trigger indicators for shop managers and customers, offering “on board sensing and communications” for food. This Use Case will be supported by third parties. We take into account diverse, numerous and powerful cyber attacks.

Thus the confrontation in SerIoT of the physical world with issues of Cybersecurity, creates a rich opportunity to move forward from traditional work in this area that focuses on cryptography and the management of cryptographic keys [71–74], or the security of software [75] and physical structures [76,77], to broad issues regarding security and system efficiency in the presence of cyberattacks to the integrated cyber and physical infrastructure.

4. Secure Routing

One of the objectives of the project is to design, implement and test a secure network infrastructure for the IoT, based on Software Defined Networks (SDN) and a smart SDN-Controller with online cognitive security surveillance and reporting, and with the ability to establish and dynamically modify paths to enhance security for IoT devices and end users, while offering a quasi-optimal level of quality of service (QoS) within the required security constraints. The online cognitive surveillance and path management of the network is based on the CPN (Cognitive Packet Network) principle that was discussed in [78] and detailed in [79]. CPNs routing level implementation and performance is presented in various papers such as [55,63]. CPN’s implementation as an overlay network is discussed in [62], and its use as a software defined network (SDN) is discussed in [80].

The smart SDN network designated “SerIoT CPN network” or SerCPN that we will develop in the SerIoT project, starts with some designs discussed in [62,80].

4.1. The SerCPN within the SerIoT Project

SerCPN is a general secure, QoS aware and energy-aware network solution, suitable for use in various application contexts, and in particular for the IoT domain, such as:

1. **Virtual network over operator’s backbone.** Telecommunication companies are starting to offer IoT solutions for their customers. The IoT network can be a separate part of the backbone network, managed by the use of virtualization techniques. SerCPN will offer SerIoT solutions to ensure the security of many customers operating IoT networks.
2. **Overlay over the public Internet.** In this case leased links are not used, but instead the resources of the public Internet will be used to create a virtual network [62] of IoT appliances. SerCPNs dynamic and distributed approach will be useful for management of the network.
3. **Local communication within the IoT network.** Some IoT networks may be very large and dynamic (e.g. vehicular networks). In such cases, the use of SDN and SerCPN [80] offer a way forward for a number of security and QoS issues.

Every node taking part in data exchange within SerCPN should be cryptographically verified, and communication should be secured. Therefore the PKI infrastructure will be implemented for authorization of the parties exchanging information and encoding of transmitted data.

SerCPN should be easily scalable to thousands of edge nodes. To achieve this scalability without loss of performance SerCPN should be decomposable into sub-networks, and coould use a distributed controller architecture, also allowing for hierarchical network structures based on Software Defined Networks (SDN) [81,82] and possible also distributed SDN [83] and hierarchical SDN [84].

4.2. Design goals

SerCPN utilises and extends the classic SDN approach. It has an architecture of separated data and control planes, and uses the OpenFlow protocol for communication between them.

Decisions about the routes the packets move are made according to the following criteria, ordered by priority:

1. **Security and Safety.** Data must be delivered in reliable way, minimizing the risk of being lost (due to intended attack or accidental failure) or intercepted. This includes protection against

hi-jacking the switches and the controller, and against attempts to feed the controller (or other network components) with wrong information.

2. **Quality of Service**. An important criterion for choosing the paths for packet delivery are the QoS parameters such as throughput, delay and jitter and this can be carried out using CPN as well [85], possible using Composite Goal Functions (CGF) [85] that include both security, QoS and possibly energy as indicated below. Indeed, SerCPN must be secure but its QoS must be attractive to customers.
3. **Energy usage**. Energy usage will be taken into account when deciding about packet routes. Using the known nodes characteristic of energy usage as a function of the load, the load of switches will be adjusted to minimise energy usage, while traffic will be distributed on paths with a view to minimizing the energy consumption per packet or per connection; this can be achieved either by using a heuristic based on CPN [86] or by a computed optimization solution as in [87].

5. Architecture of the SerCPN network

SerCPN's components include:

1. SerCPN Forwarding Element (SFE),
2. SerCPN Controller(s),
3. SerIoT Analytics Nodes will exploit data collected by SerCPN,
4. SerIoT Honeypots will also attempt to attract attacks and inform SerCPN about the network state.

SerCPN's Forwarding Element (SFE) is a basic component of the network. It is a Network Forwarding Element (NFE, referred often imprecisely as SDN switch or SDN router) modified for the needs of SerCPN. SFE performs normal NFE tasks - switching of SDN flows according to rules sent from a controller with use of the OpenFlow protocol. In addition, SFE does the task related to gathering security, QoS and energy usage data, which cannot be done by the SerIoT controller alone. SFE may have the client devices connected (edge node/edge SFE) or may be an intermediate node This piece of open-source software we will extend to achieve required features and functionalities is Open vSwitch (https://www.openvswitch.org/)

The SerCPN controller (SerCon) will be based on an advanced, up-to-date, open-source SDN controller. Consistent, modular and easy to extend internal architecture, along with required features – ability to work in distributed mode – caused that we chose ONOS (https://onosproject.org/) as the basis for SerCon. Controllers in SerCPN will be distributed or at least mirrored (with ability to take over tasks of disabled host), to avoid having single point of failure.

The aim of the **SerIoT Analytics** module (SAM) is to provide evaluation of flows in the SerCPN by statistical comparison with historical data. Flows having characteristics different than expected may be blocked, or directed towards a Honeypot, or put under observation for a later decision. The number, placement and interfaces of SAMs will be discussed with partners responsible for their implementation in the project.

The SerIoT Honeypot (SH) is a system which mimics the functions of certain devices; it is connected to SerCPN and analyzes the attacks that are conducted on itself. It can be taken over by an attacker without any harm to other nodes of SerCPN. The number, placement and interfaces of SHs to be discussed with partners responsible for their implementation in the project.

The Analytics and Honeypot modules will be developed in WP4 and WP5. Their interfaces will be agreed between IITIS (responsible for WP3) and partners responsible for the specific work packages and tasks that deal with these components of the SerIoT project. Further details of the of analytics and about honeypots are out of scope of this paper which focuses on the SerCPN architecture.

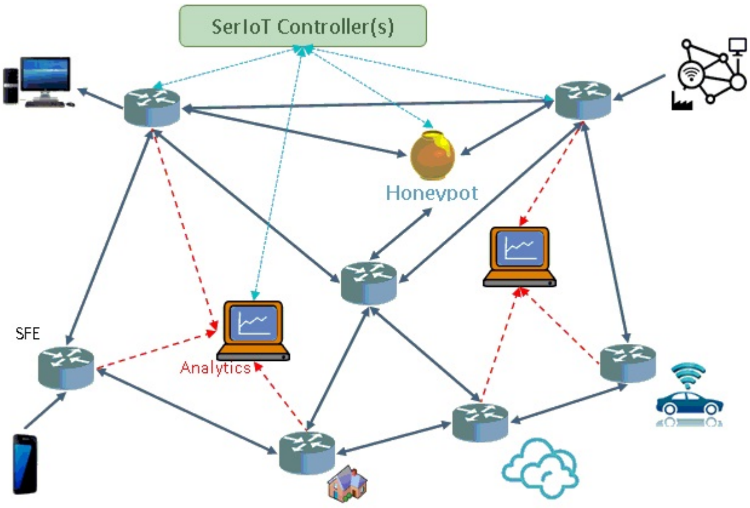

Figure 1. An example of a basic SerCPN network (single domain)

5.1. Types of SFEs

Looking at use cases, which include both stationary and mobile nodes, with different requirements (from simple messages to continuous video streaming), we can distinguish the following types of SerCPN Forwarding Elements:

1. **Intermediate nodes**, connected only to other SFEs; intermediate nodes are responsible of forwarding packets according to rules sent by SerCon and of providing data to SerCon, by direct communication or via SPs. Special case of intermediate node is **border node**, connected with SFE belonging to another subnet.
2. **Edge nodes**; every edge node does the same tasks as the intermediate node; in addition it contains a prevention module, doing simple checking of flows against predefined conditions; edge nodes may be:
    - **stationary**, having one or more cable or wireless connection,
    - **mobile**, having one or more wireless connections; main characteristics of mobile node is rapidly changing QoS and high probability of temporary connection losing.

Every edge SFE may be a:

- **Full SerCPN node** which has more than one physical communication channel, such as cable or wireless for stationary SFEs; wireless channels only in case of mobile SFEs, usually cellular and/or satellite. It may use several physical channels, e.g., GPRS/LTE and satellite, or GPRS to different operators, with distinct SIM cards and transceivers,
- **Limited SerCPN node** which could be a lightweight mobile node, implementing SerIoT features (SPs, prevention), but connected with the core network only through a single Virtual Private Network (VPN) connection, thus doing no routing. Optionally, the VPN can be dynamically reconfigurable.

In the SerIoT use cases, full mobile nodes will be installed in buses having high QoS demands (e.g., live video surveillance, plus data from sensors). Changing the wireless interface when current connection conditions are degraded will help to maintain the required QoS level. In the case of wearable sensors or simple monitoring of temperature in a food delivery van, the capabilities of a limited node will be suffice, since such a node is smaller, lighter and less energy consuming.

5.2. Hierarchical architecture of SerCPN

Due to required size of SerCPN, introduction of hierarchical architecture is necessary. The SerCPN is divided into domains, called subnets. Every subnet constitutes autonomic SDN network, controlled by First Level SerCon as shown in Figure 2. The First Level SerCon is responsible for the routing within the subnet, thanks to data gathered from its SFEs. It will also be able to route packets to neighbouring subnets (via the appropriate border node). In the case of flows having destinations outside their own subnet and neighbouring subnets, routing requests will be sent to a Second Level controller.

Second (and higher) level SerCPN controllers will see the lower level subnet as a single meta-node. They will be able to designate routing paths via meta-nodes that they control. Security and QoS meta-data are gathered via lower level SerCons. The paths through subnets, found by higher level SerCons are called meta-flows. The detailed path within a subnet is found by First Level SerCons.

A flat structure of domains is also possible in some cases for middle-scale networks. Details of data exchange between SerCons connected in a flat structure will be further discussed and investigated in future work.

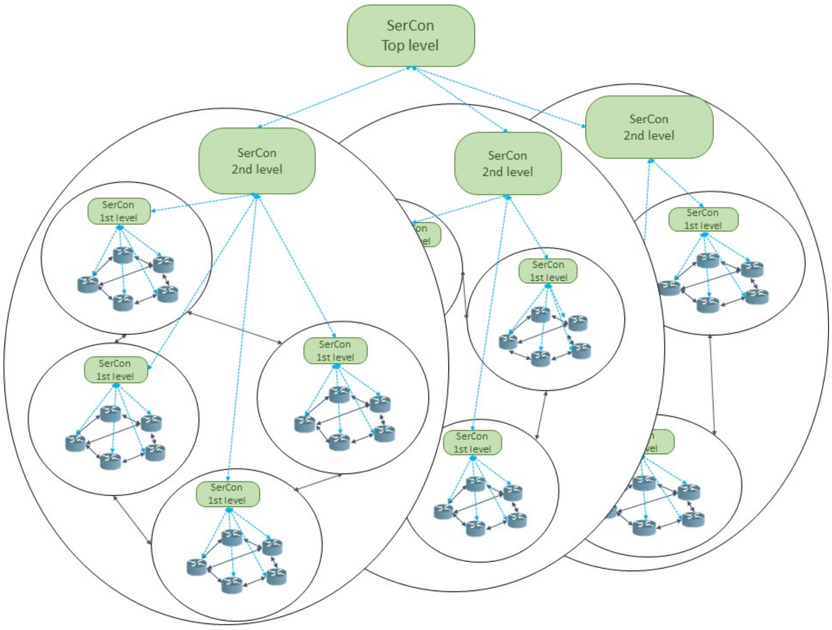

Figure 2. The Hierarchical Architecture of the SerCPN

6. Routing Criteria

In the case of a SDN based implementation for SerCPN, routing decisions are reflected by creating appropriate rules for given flows. The routing decisions will be taken by an “oracle”, which is fed with security, QoS and energy data, which will be stored in a Cognitive Security Memory (CM). As in CPN [78][79] the “oracle” will be implemented using Random Neural Networks (RNN) [88][89] which will specifically exploit a real-time learning algorithm such as Reinforcement Learning (RL). RNNs will be placed in the controller plugin, and the CM data will be used in the RNN learning process. Let us note that the RL based learning process of CPN not only exploits the ongoing measurement data, but it also generates data such as the comparison between short or long-term history regarding measurements, and the QoS, Energy or Security state of paths [85] that may be exploited by the analytics modules of SerCPN.

CM data will belong to three groups, according to the criteria listed in 4.2. A detailed list of used values, as well as their meaning for final evaluation of paths and intermediate nodes will be worked out during the timeline of the project. We will evaluate the:

1. Security (trust level) of devices connected to given edge SFE – likeliness of being the source or destination of the attack.
2. Security (trust level) of the SFEs, both edge and intermediate – likeliness of the node being attacked and disabled or intercepted.
3. Security (trust level) of particular flows – likeliness of being part of attack.
4. QoS parameters provided by particular paths.
5. Power consumption and energy usage of particular nodes for specific measured traffic values.

In all cases we will focus on, and obtain, quantitative metrics and evaluations, and these values will be used for learning by the RNNs, and transferred for exploitation by the analytics modules of SerIoT.

Group 1 includes preventive actions. Simple verifications may include the verification of default credentials, which are the biggest IoT threats due to devices having set default passwords, also the fingerprinting the firmware version (with comparisons vis-a-vis of a database containing known vulnerabilities of particular firmware versions), and more advanced – automatic active penetration testing of connected devices (one time or periodic).

Group 2 actions depend on the details of the implementation of a given network. If all SFEs are not administered by the same owner, actions similar to those regarding Group 1 should be used. In the opposite case, when all nodes are under the administration of single unit, some similar actions, e.g. pointing at the necessity of firmware upgrades, should be considered. In addition, we may use the availability rate of the nodes, independently of the reasons such as attacks or technical failure, as a way to modify the trust level of a node. More advanced methods can include the ability to learn the IP addresses and ports used by IoT devices connected to a given edge node, and – after some time of observation (e.g. a week) – firewall rules can be set to block or report traffic that does not match the known addresses or ports.

The widest group of security measures will be taken as part of **Group 3**, requiring an estimate of the probability that a given flow is part of network attack. Some of the verifications that may be used include:

- Checking if the source or destination of the flow is on the public blacklist of IP addresses known to be sources of attack.
- Checking if a DNS name appearing in a DNS request of a customer appliance is on the public blacklist of domains known to be sources of attack.
- Introducing simple (lightweight) prevention without using statistics, e.g.:
  - Detection of very high bitrate (exceeding predefined threshold)
  - Detection of access to random IP addresses (IP scan)
  - Detection of non-standard use of protocols (ssh, DNS, NTP etc.)
  - Spam detection (e-mails sent from devices we don't expect doing so, or in amounts exceeding expected numbers)

We should also consider gathering information about whether a given node (SFE or client device) is often the aim of unsuccessful attacks. Such data may be useful for evaluating its trust level.

The next group, **Group 4**, considers criteria for routing decisions based on QoS parameters. Throughput of given links, delay and jitter, loss rates in case of wireless links should be measured and forwarded to the CM.

Group 5 includes data on power consumption and energy usage. Since energy awareness is not a of primary concern to WP3, we will not focus on this point but keep it in mind for further work.

Thus the factors listed in Group 1 through Group 4 will be used to create the CGF [85] for SerCPN optimisation.

The data listed above are directly used by 1st level SerCon(s). Data gathered by the higher level SerCon(s) will be meta-data describing the subnet as a whole. Its detailed content will be worked out during the course of the project. Still, the higher level SerCon(s) will use the same concept of RNNs and CSMs, carrying out meta-routing decisions in the same manner and according to the same principles.

7. Implementation of data acquisition and routing decisions

The SerCPN Routing Engine (SRE) will be distributed in one or more SDN controllers as a plugin module, using the RNNs to implement the decision oracles, enabling a semi-distributed way of taking decisions, but using the advantages of the semi-centralisation of the SDN architecture. Linking specific RNNs to SFEs will reflect the physical network topology. The role of a single RNN will be to specify, at the time of decision making, which output node should be used for a given SFE, regarding a flow having a given destination. Data will be gathered by the SRE in three ways:

1. Upon request from the controller,
2. As unsolicited (asynchronous) messages from SFE to controllers, and
3. In the form of Smart Packets (SPs).

Requests from controller may be implemented in OpenFlow or other standard protocols, e.g. SNMP. The same applies to asynchronous messages from the SFEs.

The Cognitive Packet Network (CP) [55] uses Smart Packets (SPs) that travel from one node to another towards their destination, gathering measured data that is provided by the nodes that are visited, and the data is returned to the source node that generated the SP in ACK (acknowledgement packets) that use the inverse of the route that was effectively used by the SPs. Normally, the path of the SPs is provided by the SFEs of the nodes visited by the SPs, and the path of ACKs is source routed from the destination node back to the source. Thus in CPN [85] each of the nodes visited by a SP is able to receive and copy from the corresponding ACK, the data that was collected by a SP on its path. Such nodes can then store and exploit the data that has been collected by each SP that visited the node. On the other hand, in SerCPN, sending of SPs and routing over the network is controlled by the SREs, each of which have a CDE.

SPs will be used for data which is not available otherwise, such as delay on the link or total delay between two adjacent nodes including the delay inside nodes, and for data which can be sent by nodes directly (asynchronously or by request) but which are less urgent (e.g. energy usage). SPs aggregate the data from many nodes on their path and send them back in a single ACK message, reducing the potential communication overhead. The use of SPs could be limited to first-level SerCons. Controllers of higher level meta-networks could then use data provide by lower level controllers.

7.1. Implementation of SPs

Introducing SPs into the SDN network requires extending of Open vSwitch implementation, as in [80], and is the main difference between SFE and a standard SDN switch. SPs do not exist in any standard protocol handled by Open vSwitch, so their implementation will be part of our work. SPs must be recognised and handled in an appropriate way, and sent by SFEs, which may be achieved by standard OpenFlow commands. The node at the end of SP route sends an ACK back to the corresponding controller, which may be carried out with standard OF commands. However, in every node that is visited, new data is added to the SP so that mechanisms to recognise SPs and handle them according to SerCPN requirements will be implemented.

7.2. SRE: The SerCPN Routing Engine

The SRE will be implemented as an ONOS plugin, and will include the RNNs. The approach will combine advantages of the centralised and distributed approaches. The SFEs have access to their own share of data about the network, gathered in their CM for short-term real-time decision making. Some of this data may be transformed into historical data that is shared remotely with the Analytical Engines in the system, or with other SFEs. Thus a section of the SFE's CM may also store data regarding remote parts of the network, perhaps on a longer-term basis. Decisions about routing in a particular SFEs are taken for each node by its own RNN. Thus, bad decision will not affect the whole network, but only a local part of network, surrounding the affected node which will temporarily provide sub-optimal routing decisions.

doi:10.20944/preprints201808.0237.v1

Thus our approach is to enable a distributed architecture for SerCPN, with many domains and controllers which may be organized either in a flat structure, or in a hierarchical structure, or both, operating at different areas and time scales.

A lightweight prevention module (LPM) will be implemented for SFE. It may be part (extension) of Open vSwitch or it may be a standalone Linux kernel module interfaced with OvS. Data from this module may be sent to the controller directly or via SPs.

7.3. Modelling studies

It is common practice while designing a communication network to use queueing network models to provide a preliminary evaluation of the consequences of design choices and alternatives [90,91]. The topology of the queueing network reflects the physical or logical structure of the real network and the nodes of the network are modelled by service stations with queues corresponding to the packet buffers at nodes, waiting to be sent toward their destination. Service times correspond to the time needed to send a packet and are proportional to packet length, and will also incorporate any packet processing time at the nodes. Such models help to evaluate the random part of the transmission delays introduced by network traffic and congestion, in addition to the propagation delays. This evaluation process will also be carried out during the design of SerCPN.

We will consider both discrete event simulation models and analytical queueing models that will reflect the specific features of this network. Simulation may be more detailed but it is more time consuming, especially in the case of transient analysis where the simulation must be repeated thousands of times to obtain statistically credible distributions as a function of time. We may also adapt existing and well documented simulation tools such as OMNeT++ [92] or ns-3 [93].

When analytical methods are considered, we see in particular the potential use of diffusion approximation [94] where the stochastic process representing the queue size is replaced by diffusion process, and its probability density function serves as an approximation of the queue length distribution. The features that are in favour of the diffusion approximation are:

- The diffusion model of a single server assumes general interarrival and service time distributions, going beyond “Markov-exponential” models,
- Mathematical network models may have any topology, including a hierarchical topology, and are scalable to any number of nodes,
- The results are obtained in form of queue length and waiting time distributions, that make it easier to analyse the QoS of paths, e.g. jitter. An alternative method, the fluid -flow approximation, which is frequently used for large topologies and transient analysis, delivers only mean performance parameters and introduces larger errors but may also be considered,
- Easy separation (decomposability) of each node within a network model. Indeed, the interactions among nodes are reduced to computation of the input flow parameters at each node [95],
- The model may include “classes” of packets that have distinct routing probabilities and distinct characteristics (such as length and processing time at nodes, e.g., allowing to separate the behaviour of smart packets (SPs) and dumb or payload data packets. For instance, a distinct mathematical model may be used for SPs due to the “search” type behaviour that is imposed on their routing schemes [21,96].
- Natural ease to analyse transient states based on the solution of diffusion equations,
- The transient model can be solved step-by-step, in small time intervals, with parameters specific to these intervals; any decision of smart SDN controller concerning the dynamic routing of packets, as well as changes in the flows due to attacks and control mechanisms, may be easily reflected in time-dependent and state-dependent diffusion parameters and time-depended routing probabilities in the equations that model the flows.

In addition, numerical problems and development of software related to the solution of these large time-dependent models cn be efficiently handled [91,97]. Models of specific control aspects, such as energy-aware routing, can also considered using classical Markov models and their extensions to G-Networks [87,98].

Acknowledgements

The authors gratefully acknowledge the support of the SerIoT Research and Innovation Action, funded by the European Commission under the H2020-IOT-2016-2017 (H2020-IOT-2017) Program through Grant Agreement 780139. This paper presents the views of the authors and does not engage the views of the European Commission. The authors thank Erol Gelenbe for his contributions to an earlier version of this paper.

References

1. Gelenbe, E.; Caseau, Y. The impact of information technology on energy consumption and carbon emissions. *Ubiquity* **2015**, *2015*, 1.
2. Jiang, H.; Liu, F.; Thulasiram, R.K.; Gelenbe, E. Guest Editorial: Special Issue on Green Pervasive and Ubiquitous Systems. *IEEE Systems Journal* **2017**, *11*, 806–812. doi:10.1109/JSYST.2017.2673218.
3. François, F.; Abdelrahman, O.H.; Gelenbe, E. Impact of Signaling Storms on Energy Consumption and Latency of LTE User Equipment. 17th IEEE International Conference on High Performance Computing and Communications, HPCC 2015, 7th IEEE International Symposium on Cyberspace Safety and Security, CSS 2015, and 12th IEEE International Conference on Embedded Software and Systems, ICESS 2015, New York, NY, USA, August 24-26, 2015, 2015, pp. 1248–1255. doi:10.1109/HPCC-CSS-ICESS.2015.84.
4. Domanska, J.; Gelenbe, E.; Czachorski, T.; Drosou, A.; Tzovaras, D. Research and Innovation Action for the Security of the Internet of Things: The SerIoT Project. Recent Cybersecurity Research in Europe, EURO Cybersec 2018; Gelenbe, E.; Campegiani, P.; Czachorski, T.; Katsikas, S.; Komnios, I.; Romano, L.; Tzovaras, D., Eds. Lecture Notes CCIS No. 821, Springer Verlag, 2018.
5. Gelenbe, E. Dealing with software viruses: a biological paradigm. *information security technical report* **2007**, *12*, 242–250.
6. Gelenbe, E.; Loukas, G. A self-aware approach to denial of service defence. *Computer Networks* **2007**, *51*, 1299–1314.
7. Gelenbe, E.; Lent, R.; Nunez, A. Self-aware networks and QoS. *Proceedings of the IEEE* **2004**, *92*, 1478–1489.
8. Oke, G.; Loukas, G.; Gelenbe, E. Detecting denial of service attacks with bayesian classifiers and the random neural network. Fuzzy Systems Conference, 2007. FUZZ-IEEE 2007. IEEE International. IEEE, 2007, pp. 1–6.
9. Sakellari, G.; Hey, L.; Gelenbe, E. Adaptability and failure resilience of the cognitive packet network. *DemoSession of the 27th IEEE Conference on Computer Communications (INFOCOM2008), Phoenix, Arizona, USA* **2008**.
10. Sakellari, G.; Gelenbe, E. Adaptive resilience of the cognitive packet network in the presence of network worms. Proceedings of the NATO Symposium on C3I for Crisis, Emergency and Consequence Management, 2009, pp. 11–12.
11. Sakellari, G.; Gelenbe, E. Demonstrating cognitive packet network resilience to worm attacks. Proceedings of the 17th ACM conference on Computer and communications security. ACM, 2010, pp. 636–638.
12. Gelenbe, E.; Görbil, G.; Tzovaras, D.; Liebergeld, S.; Garcia, D.; Baltatu, M.; Lyberopoulos, G. NEMESYS: Enhanced network security for seamless service provisioning in the smart mobile ecosystem. In *Information Sciences and Systems 2013*; Springer International Publishing, 2013; pp. 369–378.
13. Gelenbe, E.; Gorbil, G.; Tzovaras, D.; Liebergeld, S.; Garcia, D.; Baltatu, M.; Lyberopoulos, G. Security for smart mobile networks: The NEMESYS approach. Privacy and Security in Mobile Systems (PRISMS), 2013 International Conference on. IEEE, 2013, pp. 1–8.
14. Abdelrahman, O.H.; Gelenbe, E.; Görbil, G.; Oklander, B. Mobile network anomaly detection and mitigation: The NEMESYS approach. In *Information Sciences and Systems 2013*; Springer International Publishing, 2013; pp. 429–438.
15. Görbil, G.; Abdelrahman, O.H.; Gelenbe, E. Storms in mobile networks. Q2SWinet’14, Proceedings of the 10th ACM Symposium on QoS and Security for Wireless and Mobile Networks, Montreal, QC,

Canada, September 21-22, 2014; Mueller, P.; Foschini, L.; Yu, R., Eds. ACM, 2014, pp. 119–126. doi:10.1145/2642687.2642688.

16. Abdelrahman, O.H.; Gelenbe, E. Signalling storms in 3G mobile networks. IEEE International Conference on Communications, ICC 2014, Sydney, Australia, June 10-14, 2014. IEEE, 2014, pp. 1017–1022. doi:10.1109/ICC.2014.6883453.
17. Desmet, A.; Gelenbe, E. Graph and analytical models for emergency evacuation. 2013 IEEE International Conference on Pervasive Computing and Communications Workshops, PERCOM 2013 Workshops, San Diego, CA, USA, March 18-22, 2013, 2013, pp. 523–527. doi:10.1109/PerComW.2013.6529552.
18. Gelenbe, E.; Wu, F.J. Future research on cyber-physical emergency management systems. *Future Internet* **2013**, *5*, 336–354.
19. Gelenbe, E. Steady-state solution of probabilistic gene regulatory networks. *Physical Review E* **2007**, *76*, 031903.
20. Gelenbe, E. Search in unknown random environments. *Physical Review E* **2010**, *82*, 061112.
21. Gelenbe, E. A diffusion model for packet travel time in a random multihop medium. *ACM Transactions on Sensor Networks (TOSN)* **2007**, *3*, 10.
22. Akinwande, O.J.; Bi, H.; Gelenbe, E. Managing crowds in hazards with dynamic grouping. *IEEE Access* **2015**, *3*, 1060–1070.
23. Gelenbe, E.; Campegiani, P.; Czachórski, T.; Katsikas, S.; Komnios, I.; Romano, L.; Tzovaras, D., Eds. *Proceedings of the 2018 ISCIS Security Workshop: Recent Cybersecurity Research in Europe*, Vol. 821. Lecture Notes CCIS, Springer Verlag, 2018.
24. Gorbil, G.; Gelenbe, E. Opportunistic communications for emergency support systems. *Procedia Computer Science* **2011**, *5*, 39–47.
25. Geneiatakis, D.; Baldini, G.; Fovino, I.N.; Vakalis, I. Towards a mobile malware detection framework with the support of machine learning. Recent Cybersecurity Research in Europe: Proceedings of the 2018 ISCIS Security Workshop, Imperial College London; Gelenbe, E.; Campegiani, P.; Czachorski, T.; Katsikas, S.; Komnios, I.; Romano, L.; Tzovaras, D., Eds. Lecture Notes CCIS No. 821, Springer Verlag, 2018.
26. Pavloski, M. Signalling Attacks in Mobile Telephony. Recent Cybersecurity Research in Europe: Proceedings of the 2018 ISCIS Security Workshop, Imperial College London; Gelenbe, E.; Campegiani, P.; Czachorski, T.; Katsikas, S.; Komnios, I.; Romano, L.; Tzovaras, D., Eds. Lecture Notes CCIS No. 821, Springer Verlag, 2018.
27. Gorbil, G.; Abdelrahman, A.H.; Pavloski, M.; Gelenbe, E. Modeling and analysis of RRC-based signaling storms in 3G networks. *IEEE Transactions on Emerging Topics in Computing* **2016**, *4*, 113–127.
28. Pavloski, M.; Gelenbe, E. Signaling Attacks in Mobile Telephony. SECRYPT 2014 - Proceedings of the 11th International Conference on Security and Cryptography, Vienna, Austria, 28-30 August, 2014, 2014, pp. 206–212. doi:10.5220/0005019802060212.
29. Pavloski, M.; Gelenbe, E. Mitigating for Signalling Attacks in UMTS Networks. In *Information Sciences and Systems 2014*; Springer International Publishing, 2014; pp. 159–165.
30. Gelenbe, E.; Pujolle, G. *Introduction aux réseaux de files d'attente*; Edition Hommes et Techniques et Techniques, Eyrolles, 1982.
31. Czachórski, T.; Gelenbe, E.; Lent, R. *Information Sciences and Systems 2014*; Springer International Publishing, 2014.
32. Staffa, M.; Coppolino, L.; Sgaglione, L.; Gelenbe, E.; Komnios, I.; Grivas, E.; Stan, O.; Castaldo, L. KONFIDO: An OpenNCP-based Secure eHealth Data Exchange System. Recent Cybersecurity Research in Europe: Proceedings of the 2018 ISCIS Security Workshop, Imperial College London; Gelenbe, E.; Campegiani, P.; Czachorski, T.; Katsikas, S.; Komnios, I.; Romano, L.; Tzovaras, D., Eds. Lecture Notes CCIS No. 821, Springer Verlag, 2018.
33. Faiella, G.; Komnios, I.; Voss-Knude, M.; Cano, I.; Duquenoy, P.; Nalin, M.; Baroni, I.; Matrisciano, F.; Clemente, F. Building an ethical framework for cross-border applications: the KONFIDO Project. Recent Cybersecurity Research in Europe: Proceedings of the 2018 ISCIS Security Workshop, Imperial College London; Gelenbe, E.; Campegiani, P.; Czachorski, T.; Katsikas, S.; Komnios, I.; Romano, L.; Tzovaras, D., Eds. Lecture Notes CCIS No. 821, Springer Verlag, 2018.
34. Akriotou, M.; Mesaritakis, C.; Grivas, E.; Chaintoutis, C.; Fragkos, A.; Syvridis, D. Random number generation from a secure photonic physical unclonable hardware module. Recent Cybersecurity Research

in Europe: Proceedings of the 2018 ISCIS Security Workshop, Imperial College London; Gelenbe, E.; Campegiani, P.; Czachorski, T.; Katsikas, S.; Komnios, I.; Romano, L.; Tzovaras, D., Eds. Lecture Notes CCIS No. 821, Springer Verlag, 2018.

35. Castaldo, L.; Cinque, V. Blockchain based logging for the cross-border exchange of eHealth data in Europe. Recent Cybersecurity Research in Europe: Proceedings of the 2018 ISCIS Security Workshop, Imperial College London; Gelenbe, E.; Campegiani, P.; Czachorski, T.; Katsikas, S.; Komnios, I.; Romano, L.; Tzovaras, D., Eds. Lecture Notes CCIS No. 821, Springer Verlag, 2018.
36. Collen, A.; Nijdam, N.A.; Augusto-Gonzalez, J.; Katsikas, S.K.; Giannoutakis, K.M.; Spathoulas, G.; Gelenbe, E.; Votis, K.; Tzovaras, D.; Ghavami, N.; Volkamer, M.; Haller, P.; Sánchez, A.; Dimas, M. GHOST - Safe-Guarding Home IoT Environments with Personalised Real-time Risk Control. Recent Cybersecurity Research in Europe: Proceedings of the 2018 ISCIS Security Workshop, Imperial College London; Gelenbe, E.; Campegiani, P.; Czachorski, T.; Katsikas, S.; Komnios, I.; Romano, L.; Tzovaras, D., Eds. Lecture Notes CCIS No. 821, Springer Verlag, 2018.
37. Horváth, M.; Buttyán, L. Problem Domain Analysis of IoT-Driven Secure Data Markets. Recent Cybersecurity Research in Europe: Proceedings of the 2018 ISCIS Security Workshop, Imperial College London; Gelenbe, E.; Campegiani, P.; Czachorski, T.; Katsikas, S.; Komnios, I.; Romano, L.; Tzovaras, D., Eds. Lecture Notes CCIS No. 821, Springer Verlag, 2018.
38. Brun, O.; Yin, Y.; Gelenbe, E.; Kadioglu, Y.M.; Augusto-Gonzalez, J.; Ramos, M. Deep Learning with Dense Random Neural Networks for Detecting Attacks against IoT-connected Home Environments. Recent Cybersecurity Research in Europe: Proceedings of the 2018 ISCIS Security Workshop, Imperial College London; Gelenbe, E.; Campegiani, P.; Czachorski, T.; Katsikas, S.; Komnios, I.; Romano, L.; Tzovaras, D., Eds. Lecture Notes CCIS No. 821, Springer Verlag, 2018.
39. Gelenbe, E.; Kadioglu, Y.M. Energy Life-Time of Wireless Nodes with Network Attacks and Mitigation. Proceedings of ICC 2018, 20-24 May 2018, W04: IEEE Workshop on Energy Harvesting Wireless Communications. IEEE, 2018.
40. Gelenbe, E.; Yin, Y. Deep learning with random neural networks. Neural Networks (IJCNN), 2016 International Joint Conference on. IEEE, 2016. doi:10.1109/IJCNN.2016.7727393.
41. Gelenbe, E. Réseaux neuronaux aléatoires stables. *Comptes Rendus de l'Académie des sciences. Série 2* **1990**, *310*, 177–180.
42. Gelenbe, E. Learning in the recurrent random neural network. *Neural Computation* **1993**, *5*, 154–164.
43. Gelenbe, E.; Cramer, C. Oscillatory corticothalamic response to somatosensory input. *Biosystems* **1998**, *48*, 67–75.
44. Cramer, C.E.; Gelenbe, E. Video quality and traffic QoS in learning-based subsampled and receiver-interpolated video sequences. *IEEE Journal on Selected Areas in Communications* **2000**, *18*, 150–167.
45. Kouzinopoulos, C.S.; Spathoulas, G.; Giannoutakis, K.M.; Votis, K.; Pandey, P.; Tzovaras, D.; Katsikas, S.K.; Collen, A.; Nijdam, N.A. Using Blockchains to strengthen the security of Internet of Things. Recent Cybersecurity Research in Europe: Proceedings of the 2018 ISCIS Security Workshop, Imperial College London; Gelenbe, E.; Campegiani, P.; Czachorski, T.; Katsikas, S.; Komnios, I.; Romano, L.; Tzovaras, D., Eds. Lecture Notes CCIS No. 821, Springer Verlag, 2018.
46. Gelenbe, E.; Domanska, J.; Czachorski, T.; Drosou, A.; Tzovaras, D. Security for Internet of Things: The SerIoT Project. International Symposium on Networks, Computers and Communications, Proceedings of the. IEEE, 2018.
47. Domanska, J.; Czachòrski, T.; Nowak, M.; Nowak, S.; Gelenbe, E. SerCPN: Smart Software Defined Network for IoT. To appear, 2018.
48. Gelenbe, E. A Software Defined Self-Aware Network: The Cognitive Packet Network. IEEE 3rd Symposium on Network Cloud Computing and Applications, NCCA 2014, Rome, Italy, February 5-7, 2014, 2014, pp. 9–14. doi:10.1109/NCCA.2014.9.
49. François, F.; Gelenbe, E. Towards a cognitive routing engine for software defined networks. 2016 IEEE International Conference on Communications, ICC 2016, Kuala Lumpur, Malaysia, May 22-27, 2016, 2016, pp. 1–6. doi:10.1109/ICC.2016.7511138.
50. François, F.; Gelenbe, E. Optimizing Secure SDN-Enabled Inter-Data Centre Overlay Networks through Cognitive Routing. 24th IEEE International Symposium on Modeling, Analysis and Simulation of

Computer and Telecommunication Systems, MASCOTS 2016, London, United Kingdom, September 19-21, 2016, 2016, pp. 283–288. doi:10.1109/MASCOTS.2016.26.

51. Gelenbe, E.; Lent, R.; Xu, Z. Measurement and performance of a cognitive packet network. *Computer Networks* **2001**, *37*, 691–701.
52. Gelenbe, E.; Lent, R.; Xu, Z. Design and performance of cognitive packet networks. *Performance Evaluation* **2001**, *46*, 155–176.
53. Gelenbe, E.; Lent, R.; Montuori, A.; Xu, Z. Cognitive packet networks: QoS and performance. Modeling, Analysis and Simulation of Computer and Telecommunications Systems, 2002. MASCOTS 2002. Proceedings. 10th IEEE International Symposium on. IEEE, 2002, pp. 3–9.
54. Gelenbe, E.; Liu, P. QoS and routing in the cognitive packet network. World of Wireless Mobile and Multimedia Networks, 2005. WoWMoM 2005. Sixth IEEE International Symposium on a. IEEE, 2005, pp. 517–521.
55. Gelenbe, E. Steps toward self-aware networks. *Communications of the ACM* **2009**, *52*, 66–75.
56. Gelenbe, E. Random neural networks with negative and positive signals and product form solution. *Neural Computation* **1989**, *1*, 502–510.
57. Gelenbe, E.; Lent, R.; Xu, Z. Towards networks with cognitive packets. In *Performance and QoS of next generation networking*; Springer, 2001; pp. 3–17.
58. Gelenbe, E.; Lent, R. Power-aware ad hoc cognitive packet networks. *Ad Hoc Networks* **2004**, *2*, 205–216.
59. Gelenbe, E. Genetic algorithms with analytical solution. Proceedings of the 1st annual conference on genetic programming. MIT Press, 1996, pp. 437–443.
60. Gelenbe, E.; Liu, P.; Laine, J. Genetic algorithms for route discovery. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **2006**, *36*, 1247–1254.
61. Liu, P.; Gelenbe, E. Recursive routing in the cognitive packet network. Testbeds and Research Infrastructure for the Development of Networks and Communities, 2007. TridentCom 2007. 3rd International Conference on. IEEE, 2007, pp. 1–6.
62. Brun, O.; Wang, L.; Gelenbe, E. Big data for autonomic intercontinental communications. *IEEE Transactions on Selected Areas in Communications* **2016**, *34*, 575–583.
63. Wang, L.; Gelenbe, E. Adaptive dispatching of tasks in the Cloud. *IEEE Transactions on Cloud Computing* **2018**, *6*, 33–45.
64. Wang, L.; Brun, O.; Gelenbe, E. Adaptive workload distribution for local and remote clouds. Systems, Man, and Cybernetics (SMC), 2016 IEEE International Conference on. IEEE, 2016, pp. 003984–003988.
65. Gelenbe, E.; Mahmoodi, T. Energy-aware routing in the cognitive packet network. *Energy* **2011**, pp. 7–12.
66. Gelenbe, E.; Mahmoodi, T. Distributed Energy-Aware Routing Protocol. In *Computer and Information Sciences II*; Springer London, 2012; pp. 149–154.
67. Gelenbe, E.; Sakellari, G.; D'arienzo, M. Admission of QoS aware users in a smart network. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)* **2008**, *3*, 4.
68. Gelenbe, E.; Ngai, E.C.H. Adaptive qos routing for significant events in wireless sensor networks. Mobile Ad Hoc and Sensor Systems, 2008. MASS 2008. 5th IEEE International Conference on. IEEE, 2008, pp. 410–415.
69. Ngai, E.C.; Gelenbe, E.; Humber, G. Information-aware traffic reduction for wireless sensor networks. Local Computer Networks, 2009. LCN 2009. IEEE 34th Conference on. IEEE, 2009, pp. 451–458.
70. Gelenbe, E.; Ngai, E.C.; Yadav, P. Routing of High-Priority Packets in Wireless Sensor Networks. *IEEE Second International Conference on Computer and Network Technology, IEEE* **2010**.
71. Levi, A.; Çaglayan, M.U.; Koç, Ç.K. Use of nested certificates for efficient, dynamic, and trust preserving public key infrastructure. *ACM Trans. Inf. Syst. Secur.* **2004**, *7*, 21–59. doi:10.1145/984334.984336.
72. Ermis, O.; Bahtiyar, S.; Anarim, E.; Çaglayan, M.U. A key agreement protocol with partial backward confidentiality. *Computer Networks* **2017**, *129*, 159–177. doi:10.1016/j.comnet.2017.09.008.
73. Yu, C.M.; Ni, G.K.; Chen, Y.; Gelenbe, E.; Kuo, S.Y. Top-k query result completeness verification in sensor networks. Communications Workshops (ICC), 2013 IEEE International Conference on. IEEE, 2013, pp. 1026–1030.
74. Yu, C.; Ni, G.; Chen, I.; Gelenbe, E.; Kuo, S. Top-k Query Result Completeness Verification in Tiered Sensor Networks. *IEEE Transactions on Information Forensics and Security* **2014**, *9*, 109–124. doi:10.1109/TIFS.2013.2291326.

75. Siavvas, M.; Gelenbe, E.; Kehagias, D.; Tzovaras, D. Static analysis-based approaches for secure software development. Proceedings of the 2018 ISCIS Security Workshop: Recent Cybersecurity Research in Europe; Gelenbe, E.; Campegiani, P.; Czachorski, T.; Katsikas, S.; Komnios, I.; Romano, L.; Tzovaras, D., Eds. Lecture Notes CCIS, Springer Verlag, 2018, Vol. 821.
76. Gelenbe, E.; Wu, F.J. Large scale simulation for human evacuation and rescue. *Computers & Mathematics with Applications* **2012**, *64*, 3869–3880.
77. Gelenbe, E.; Bi, H. Emergency navigation without an infrastructure. *Sensors* **2014**, *14*, 15142–15162.
78. Gelenbe, E.; Xu, Z.; Seref, E. Cognitive packet networks. Tools with Artificial Intelligence 1999. Proceedings. 11th IEEE International Conference on, 1999, pp. 47–54.
79. Gelenbe, E. Cognitive Packet Network. *US Patent 6,804,201* **2004**.
80. François, F.; Gelenbe, E. Towards a cognitive routing engine for software defined networks. ICC 2016. IEEE Xplore, 2016, pp. 1–6.
81. Jammal, M.; Singh, T.; Shami, A.; Asal, R.; Li, Y. Software defined networking: State of the art and research challenges. *Computer Networks* **2014**, *72*, 74–98.
82. Stallings, W. *Foundations of modern networking: SDN, NFV, QoE, IoT, and Cloud*; Pearson Education, 2015.
83. Bannour, F.; et al.. Distributed SDN Control: Survey, Taxonomy and Challenges. *IEEE Communications Surveys and Tutorials* **2017**.
84. Liu, Y.; et al.. A SDN-based architecture for horizontal Internet of Things services. Proc. IEEE Int. Conf. on Communications (ICC), 2015, pp. 5374–5379.
85. Gelenbe, E. Steps toward self-aware networks. *Commun. ACM* **2009**, *52*, 66–75.
86. Gelenbe, E.; Mahmoodi, T. Energy-aware routing in the cognitive packet network. ENERGY, 2011, pp. 7–12.
87. Gelenbe, E.; Morfopoulou, C. A Framework for Energy-Aware Routing in Packet Networks. *Computer Journal* **2011**, *54*, 850–859.
88. Gelenbe, E. Reseaux neuronaux aléatoires stables. *Comptes Rendus de l'Académie des Sciences. Série 2* **1990**, *310*, 177–180.
89. Gelenbe, E. Learning in the recurrent random neural network. *Neural Computation* **1993**, *5*, 154–164.
90. Gelenbe, E.; Pujolle, G. *Introduction aux Réseaux de Files d'Attente*; Editions Hommes et Techniques, Eyrolles, 1982.
91. Czachórski, T.; Pekergin, F. Diffusion Approximation as a Modelling Tool. Network Performance Engineering – A Handbook on Convergent Multi-Service Networks and Next Generation Internet; D.Kouvatsos., Ed. Springer, 2011, pp. 447– 476.
92. Webpage. OMNeT++ Discrete Event Simulator.
93. *WNS3 '16: Proceedings of the Workshop on Ns-3*, New York, NY, USA, 2016. ACM.
94. Gelenbe, E. On Approximate Computer Systems Models. *Journal ACM* **1975**, *22*, 261–269.
95. Gelenbe, E.; Pujolle, G. The Behaviour of a Single Queue in a General Queueing Network. *Acta Informatica* **1976**, *7*, 123–136.
96. Gelenbe, E. Search in unknown random environments. *Physical Review E* **2010**, *82*, 061112.
97. Czachórski, T. Queueing models for performance evaluation of computer networks, transient state analysis. Analytic Methods in Interdisciplinary Applications. Springer Proceedings in Mathematics & Statistics, 2015, Vol. 116, pp. 51–80.
98. Gelenbe, E.; Morfopoulou, C. Routing and G-Networks to Optimise Energy and Quality of Service in Packet Networks. E-Energy 2010; Hatziargyriou, N.; et al.., Eds. Springer LNICST 54, 2011, p. 163–173.