Using Programmable Data Networks to Detect Critical Infrastructure Challenges

Kyle J. S. White, Dimitrios P. Pezaros, and Chris W. Johnson

School of Computing Science, University of Glasgow
Glasgow, G12 8QQ , Scotland
mail@kylewhite.com,{dimitrios.pezaros,christopher.johnson}@glasgow.ac.uk

Abstract. *Critical infrastructures must be better protected against challenges to their data communications in the face of increasing numbers of emerging challenges, complexity and society's demand and intolerance of failures. In this paper, we present a set of challenges and their characteristics by reviewing reported incidents. Using domain specific attributes we discuss how these could be mitigated. We advocate the adoption of the latest programmable networking approaches in critical infrastructure networks and we present our proposed modular architecture with configurable monitoring and security components. Lastly, we show results from a network challenge simulation which highlights the benefits of our approach in providing rapid, precise and effective challenge detection and mitigation.*

Keywords: Resilience, Security, Critical Systems

1 Introduction

Data communication networks are playing an increasingly pivotal role in critical infrastructures. Society is becoming ever more reliant on critical infrastructures to manage a range of services including communications, power and transportation. It is therefore necessary that the underlying data networks of such infrastructures remain highly resilient and available in the face of emerging challenges. Data networks worldwide are becoming progressively interconnected and explicitly, critical infrastructures are becoming less isolated, e.g., EU Air Transportation systems are increasing their interconnectivity through an international Airport Collaborative Decision Making (A-CDM) system [2]. In this paper, we argue that critical infrastructure data networks are encountering a series of emerging challenges. We discuss the motivation for increasing data communication resilience and the domain specific attributes of critical infrastructures we can exploit, in Section 2. In Section 3, we review reported incidents within our focussed context of mission-critical Air Traffic Management (ATM) systems. By analysing these incidents and considering other possible scenarios, we have characterised a taxonomy of emerging challenges and their features. In order to best take advantage of the attributes of critical network infrastructures (CNI), we argue for the introduction of the latest programmable networking approaches, specifically Software Defined Networking (SDN). SDN allows centralised, logical

control of data traffic routing. This offers unique opportunities as event based code can be executed in response to real-time network behaviour, e.g. allowing the controller to enforce different policies depending on the traffic type. Using our domain specific knowledge and familiarity with SDN, we argue that the typical qualities of many CNIs can be exploited using tailored algorithms on SDN-based architectures that will allow them to better mitigate such challenges than would be possible in other data networks. We then present our vision of a modular architecture in Section 4 and discuss how it can be utilised to improve resilience. Our final contribution, in Section 5, is an initial experiment which highlights the benefits of our architecture, and SDN, when detecting and remediating against a flooding challenge, before concluding the paper.

2 Characteristics of Critical Network Infrastructures

Data networks within critical infrastructures have certain characteristics that hold true more predominately than for data networks in general. Firstly, they should be highly redundant with respect to both topology and services. Secondly, any changes are perceived as high risk. Finally, high availability and low latency are required. These three attributes converge neatly. To achieve high availability, there should be significant redundancy. Low latency requires over-provisioning and considerate routing strategies for queueing and congestion. To make changes to a system undermines the experienced reliability to date as unknown consequences can occur. Leaving the system unchanged is therefore perceived as the best means to maintain achieved levels of high availability. This means traffic patterns and topologies are relatively static in comparison with the ad-hoc, ever-changing behaviour of the legacy Internet. As technology is advancing and some changes are being made, such as reducing the relative isolation of critical infrastructures, it is now time to make further, protective, changes to safeguard critical infrastructure data network's resilience. As evidence for the increasing interdependency of systems we refer to the new EU networked system to directly share collaborative flight planning data among different systems and countries [2] and for a comprehensive overview of CNI's interdependencies in general, see [13].

The complexity of critical infrastructure communication networks is rising as they age. Current data networks are moving larger bandwidths of data traffic and systems are processing ever greater volumes of data. This trend continues as capacity improves and as demand increases for more knowledge of the system with more intelligent data correlations. Modern critical infrastructures often have diverse technologies integrated into the overall system. This has come from building improvements onto legacy components and adapting new technology to interface with existing protocols.

2.1 Energy and Communication Resilience

Critical infrastructures rely on power and communications in order to maintain mission-critical services. These two components are often interconnected with

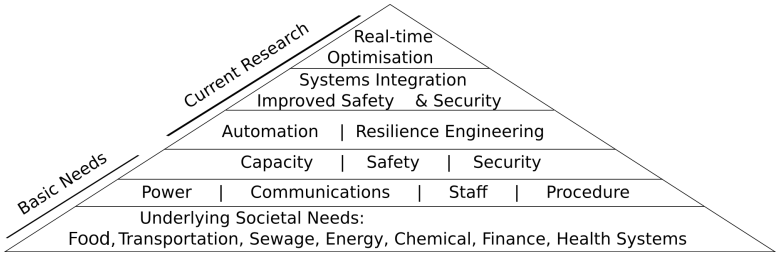

Fig. 1. Depiction of the layers of dependency in critical infrastructure systems

energy required to power communications and a communications network required to manage, monitor and control the power systems. Nearly all elements of critical infrastructures require both power and communications to perform their function. In Figure 1, we present our view of the functional hierarchy of components required to effectively provide the services which comprise critical infrastructures. The pyramid requires the underlying components to be in place to offer the services at higher layers. We differentiate between energy the CI national grid supply, and the power required for an individual CI to function internally. Capacity, safety, security, automation and resilience engineering fall between basic needs and advancing current research. Systems intergration, real-time processing and optimisation form the pinnacle of current CI research.

Given the relationship discussed above, power and communications can be categorised together. However, as we discuss in previous work [19], the challenges faced in communications resilience are greater. Power supplies are source-independent, with energy needs being met from mains, back-up generators and even fail-safe battery power in the worst case. Voltage spikes or brown outs can be mitigated using standard techniques such as Diesel Rotary Uninterruptible Power Supplies which convert the power to a steady, clean stream in terms of phase, harmonic distortion and consistent voltage. Communications resilience is a more sophisticated problem as it is time-critical for real-time applications and the content is source-dependent. Communication payloads are a complex, non-Poisson process of traffic load and arrivals. This implies high variability and unpredictable dynamics over long timescales. Traffic peaks can also be significant in terms of utilisation, even with substantial over-provisioning. Data packets are also susceptible to corruption, loss and delay. We therefore argue there is strong motivation to place a greater emphasis on communications resilience to ensure the same levels of reliability and availability as are present in power resilience.

3 Examples of CNI Challenges

Numerous challenges threaten the failure of critical infrastructures. We have defined a generalised taxonomy of characteristics for a series of accidental failures and malicious attacks which can be categorised together. We base this categorisation on the behaviour of the challenge in the data network, the characteristics

of CNIs can be exploited to defend against such challenges and the existence of past incidents which are an instance of such a challenge or allude to the scenario being a possible, legitimate threat. The challenges in this partial taxonomy are representative because they have a similar manifestation but with different implications. There are also common attributes we can use to detect these challenges which we review after presenting our taxonomy. We consider:

- Flooding:
  - Distributed Flood
  - Centralised Flood
- Disconnectivity
- Oscillation
- Network Scan

Flooding: Network flooding is where a large amount of data overwhelms the available capacity of the network resources, e.g., in terms of physical bandwidth. Other types of flooding exist such as SYN flooding, where a large number of TCP flows are simultaneously sent to a remote server initiating the connection protocol. The server sends TCP SYN-ACK and awaits ACK replies for each of these flows, before running out of memory, thus stopping new connections being made. We distinguish between two types of flooding: *distributed* and *centralised*. Distributed floods are well-known for causing Distributed Denial of Service (DDoS) attacks, Flashcrowds and Botnets. These challenges have a distributed source address space and a concentrated destination address space. Centralised floods begin from a focussed point in the network and can have a single root cause. They may however propagate, such as a case of a broadcast storm, where network devices forward data packets to all connected devices, as instructed, resulting in an endless flow of traffic. Characteristics of this type of flooding are a concentrated source address space and a concentrated destination address space. These are well known challenges and many solutions exist to mitigate them in generic network environments [10].

Disconnectivity: Disconnectivity is where part of the network infrastructure is no longer available. This may be due to hardware, e.g., a physical link or switch going down or software misconfiguration of a firewall or routing table errors. The threat of failure from a disconnectivity challenge is independent of its cause. While it is important to rapidly diagnose and isolate the cause of the problem, in terms of overall system availability, it is more important to understand the impact of the challenge. Depending on where the disconnectivity arises, this may mean an alternative path between parts of the network experiences higher traffic levels as data is rerouted around the problem area or it may mean parts of the network are entirely isolated. Characteristics of this challenge will likely be a significant drop in traffic in some parts of the infrastructure, a similarly significant rise in traffic in other parts, as well as the potential creation of additional routing paths throughout the network.

Oscillation: Link redundancy is a core feature of mission-critical networks. Even in simple topologies, persistent routing oscillations can occur. This challenge makes understanding normal behaviour complex and is bad for stability, load balancing, reliability, predictability and fault detection [4]. It also degrades router performance and makes monitoring data harder to interpret. Finally, anomaly detection methodologies are also hampered since many rely on trends

or sampling [1], and the characteristics of oscillation are bursty traffic conditions and routing table variability.

Network Scan: Network scans and other distributed attacks such as Exploits and Worms can be categorised together with common characteristics of a distributed destination address space, focussing on a limited set of destination ports. These challenges exploit network vulnerabilities and are malicious attacks. In traditional networks firewalls and network monitoring tools offer some protection against these attacks.

While distinct, there are common aspects to which can be used to optimally detect these challenges, particularly when in an SDN environment. The key aspect for this taxonomy is the desire to have both distributed and centralised knowledge to detect and mitigate the challenge. Flooding incidents require rapid identification of the source of the traffic. This can be achieved through a distributed alarms which recognise flood traffic, and centralised control which can determine if the event is localised, distributed or propagating throughout the network. Similarly, disconnectivity requires both distributed knowledge of local behaviour as well as a global perspective to determine where the cause is and whether any parts of the network are now completely isolated. Oscillation can occur at various scales or across local domains making it necessary to examine at both levels. Finally, network scans vary and depending on their nature may be obscured at either a local or global level. If a network scan is mounted in a distributed address space attacking the same port, this requires a centralised detection algorithm. However, if a port scan is targeting a concentrated destination address space and examining each of its ports, this can be handled locally.

3.1 Review of CNI Incidents

To provide evidence for our claims of the challenges facing critical infrastructures we explore recent incidents involving Air Traffic Transportation Systems. The following four incidents highlight the severity of the impact faced when challenges do undermine the resilience of ATM data networks and therefore the safety and security of the service which relies upon them.

In 2007, the US Customs and Border Protection computer systems at LAX suffered from a network outage [8]. The fault analysis concluded the single initial point of failure was caused by a malfunctioning Network Interface Card (NIC). This in turn caused data to overload the system. This system is not directly involved in ATM services however, the 10-hour-long issue caused delays and congestion affecting up to 17,000 passengers with new arrivals not being allowed to disembark, and international departure disruptions. Analysis suggests this incident had similar characteristics to a *centralised flooding* challenge which propagated throughout the network.

In 2008, a European airport ATM system experienced a similar incident caused by an intermittent faulty NIC. This fault and the subsequent error of automated network software designed to mitigate anomalous issues ultimately caused periods of complete airport closure due to safety concerns. The seven week period over which problems persisted included times where ATM lost track of

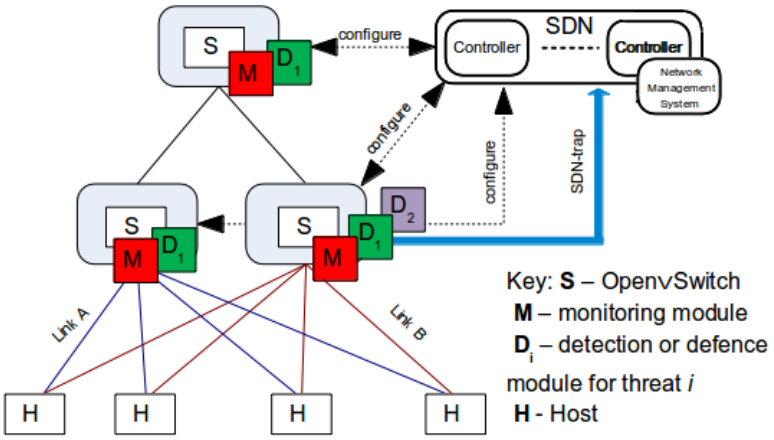

Fig. 2. Modular SDN architecture with distributed, configurable monitoring & security

planes or associated flight information [5]. This was a complex incident with multiple factors involved, however, the root cause gives further evidence to the threat faced from *flooding* challenges.

The FAA (Federal Aviation Administration) Telecommunications Infrastructure (FTI) experienced a four-hour outage in 2009 causing disruption for over 800 flights [3]. The incident was a series of cascading events which culminated in failure. Earlier scheduled maintenance led to a routing table being programmed incorrectly. This was inactive until it was restarted. Independently, alarms monitoring router utilisation had been inadvertently disabled for all routers. The lack of an alarm system compounded the routing error and a significant delay occurred while network engineers manually probed the network to localise the problem and eventually determine which router was at fault. Given a routing failure scenario such as this and the potential for human error due to manual programming, it is probable severe oscillation incidents can occur as well as significant flooding, in order to have router utilisation exceeding alarm thresholds.

Most recently, a fire caused disconnectivity at a core FAA PoP (Point of Presence) in Atlantic City in 2012 [12]. The building was evacuated and caused some air traffic and flight planning systems in the U.S. to become temporarily unavailable. The ATM service was significantly slowed. Back-up systems relied on telephones for communications. This incident was well-managed but highlights that despite high levels of redundancy, disconnectivity is a very real threat and the challenge of continuing operations seamlessly in the face of disrupted infrastructure connectivity cannot be guaranteed.

4 Proposed CNI Architecture

In order to better tackle the threat of failure that the emerging challenges we have outlined present, we argue that the latest networking technologies should

be introduced into ATM data networks. By applying SDN as the foundations for our proposed architecture, numerous advantages can be gained. We believe that by augmenting SDN functionality with our specific architecture, distributed algorithms can be tailored to exploit critical infrastructure attributes to better defend against a wide array of future challenges including those in our taxonomy. To begin, we will introduce the benefits of an SDN approach alone and then we will discuss our proposed additions which further increase resilience.

4.1 SDN for Critical Infrastructures

SDN is a networking approach which separates the data and control planes. The data plane handles the forwarding of network traffic to the correct destination space while the control plane handles the decision on how traffic should be routed. There is a single logical controller, which may be physically distributed, which sets control plane rules in the flow tables of switches. These rules match packets which arrive at the switch. If the information in the packet header matches a rule, the switch performs the associated action, e.g., forwards the packet on the data plane. If there is no match, the packet is sent to the controller which can decide to establish a new rule and install it on the switch. The controller can modify existing rules and actions can be set to discard, forward packets to their destination or send them to the controller to handle. There has been a great deal of work on various related aspects of network functionality in an SDN environment including anomaly detection techniques [9, 20], integrating different approaches to monitoring and security services [16, 17], network verification [6] and automatic failure recovery [7]. OpenFlow, a protocol specification which is layered on top of TCP, is the first standard communications interface defined between the control and forwarding layers of an SDN architecture. It relies on Open vSwitches with routing flow tables. The *POX* OpenFlow controller and others [14], allow operators to specify event-based functionality.

Of the many advantages SDN brings to a network operator, those of most specific interest to critical infrastructures include: vendor-independent centralised management and control; centralised and automated management of network devices allowing uniform policy enforcement thus reducing configuration errors and increasing network reliability and security; fine-grained control of varying services, users, devices and application policies; improved automation and management by using common APIs to manipulate the underlying network behaviour for provisioning systems and applications [11].

Crucially, SDN is vendor-independent and can therefore be used on a variety of different devices. This is advantageous since, as we discussed earlier, many of the CNIs comprise a mixture of legacy systems, conversion networking devices and differing manufacturer equipment. Further related work [15], which separates a physical network infrastructure into different logical networks could be of significant interest to critical infrastructures when it matures. SDN technology could allow changes to be test-deployed alongside operational CNIs in the same environment as a final safeguard before migration from the testbed to live deployment.

4.2 An SDN Architecture for ATM

While OpenFlow offers an excellent environment to explore some of the advantages of SDN we believe more can be achieved to aid safety and security through an SDN approach. With increasing bandwidths and data processing, anomaly detection is requiring faster processing in order to be effective in a real-time context. We propose that to achieve truly optimised, efficient anomaly detection for critical infrastructures, the processing of detection methodologies should be distributed. To achieve this, switches would require more resources, however we argue this is a realistic proposition for two reasons: the costs of computing components are reducing and with the separation of control and data planes through SDN there are far greater possibilities for distributing anomaly detection algorithms. We therefore believe the cost-benefit trade-off for this architecture is now in favour of distributed anomaly detection, particularly to exploit the environment of CNIs. In Figure 2 we show our modular approach to an SDN architecture for critical infrastructures, which has centralised knowledge and distributed intelligence. Open vSwitches are configured by an SDN controller as standard. Hosts are connected to Open vSwitches by 2-fold physically redundant links: Links A & B. The novelty of our architecture comes from the configurable monitoring and security modules. These are pre-tested modules which can be deployed on-demand by network operators to perform a task given the current status of the network. This distributed approach allows network operators to have the freedom and control over their resources to respond to challenges as they emerge. When an incident unfolds, operators want to learn more in real time. This architecture offers them the ability to do that rapidly. If a given threat is presumed to be causing a problem in the network, security detection or defence modules specific to that threat can be deployed to a given switch to gain further information and attempt to self-heal. Similarly, if detailed performance information is desired prior to, for example, network upgrades, a suite of monitoring tools can be deployed throughout the network to learn about the localised behaviour in that part of the infrastructure. Figure 2, also details an SDN-trap. We define this as an asynchronous message from the switch to the controller which could be used to alert the controller that anomalous behaviour has been identified.

5 Flood Detection and Remediation Experiment

To prove the merits of our architecture we implemented a flooding detection and remediation experiment based on challenge characteristics and our domain specific knowledge. This experiment is designed to show the advantages and potential which can be leveraged for critical infrastructures using our SDN-based architecture and early feasibility results. We simulated a scale version of a major European Air Navigation Service Provider's (ASNP) secondary radar surveillance network. In Figure 3, we show our experimental topology. The core of the network is a ring connecting switches S_1, S_2 and S_3. These switches represent the primary ATM locations throughout the country. H_1, H_2 and H_3 represent the subscribers to the surveillance data to, for example, display where aircraft are

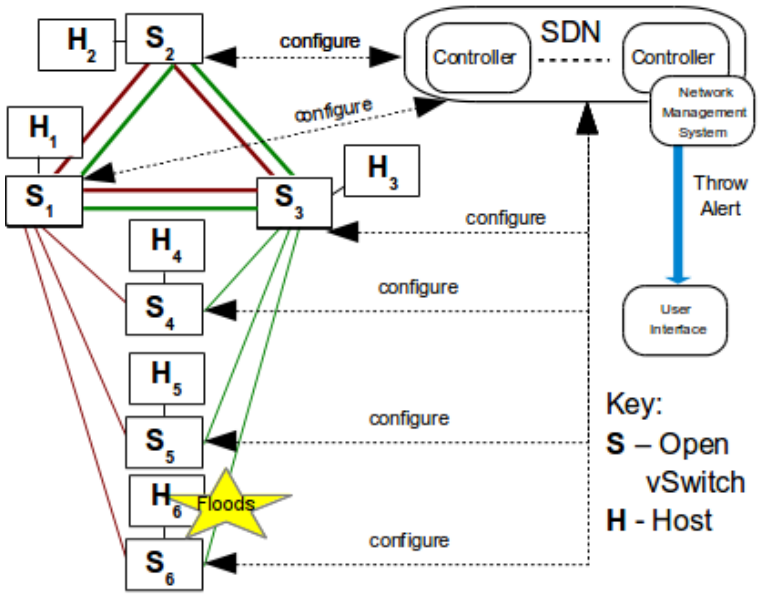

Fig. 3. Flooding experiment on scaled ANSP radar surveillance network topology

located on Air Traffic Controller displays. In reality these hosts are in fact large Local Area Networks (LANs) with their own layers of redundancy. In Figure 3, the core ring network connecting switches S_1, S_2 and S_3 has a high bandwidth and shares the captured radar data from the distributed radar locations, represented here as H_4, H_5 and H_6. Each radar dish has a local switch which sends dual copies of the output data on the Red and Green links which represent the 2-fold physical redundancy in the network. Each switch is configurable through the centralised SDN controller.

5.1 Methodology

To begin, standard operational traffic was initiated in the network with H_4, H_5 and H_6 sending their continual radar data to the core ring via the Red and Green links to S_1 and S_3, respectively. The operational traffic was modelled from recordings of live data which we analysed in our previous work [19]. As the standard traffic began, the POX controller performed its default behaviour, establishing flow table entries for the switches on the core ring, allowing their associated hosts to directly route traffic to each other. Flow table rules were also installed on the radar switches S_4, S_5 and S_6 to allow them to send traffic from their hosts to H_1 and H_3.

In our controller, we implemented a traffic metric polling for each switch to provide the number of flows, the number of bytes and the number of packets sent from that switch to each destination address. This polling was triggered by a timer event every five seconds. From our knowledge of surveillance networks

and following the static characteristics of critical infrastructure data networks, we determined that new connections between pairs of devices which had never previously exchanged data is a relatively rare event in this domain. Once the network is established and the controller has installed flow table entries for standard operational traffic patterns, new connections to send data from e.g. H_4 to H_5, are unlikely. We exploited this characteristic to better detect a flooding challenge in the network. Every time a packet is sent to the controller and a new flow table entry is installed on a switch, we add this forwarding rule into a list of the latest added routes. Each time a timer event is called, we examine the number of new routes which are added in the network. If this number exceeds a given threshold for new connections made, this indicates abnormal behaviour within the network. Such behaviour could be representative of a *network scan*, *centralised flood* or *disconnectivity* challenges. By then checking for a significant increase in the volume of data sent from this switch via the polled traffic metrics, we can determine a centralised flooding incident. The algorithm used is:

```
On Timer Event:
  for each Switch in Latest Route Entries:
    if number of new routes for this Switch > MAX_NEW_ROUTES_THRESHOLD:
      if significant traffic volume increase:
        Throw Flood Characteristics Alert
      else: Throw Generic Alert
```

On completion of checking the latest route entries we archive them. Any new flow table entries created in the next time period will be evaluated independently of archived results. This is based on the profile of a flooding event typically being a rapid process in which a malfunction or misconfiguration rapidly causes data to be sent from a constrained set of sources to a large set destinations. When our system throws a flood characteristic event, we pass the details of the switch, its latest routes and the traffic volumes. The controller then exposes these alerts which can be collected by a network monitoring system and reviewed through e.g. a web based user interface such as KSWatch [18]. Network operators could then act using our modular architecture to block flows from this switch, increase detection in other parts of the network and perhaps deploy further monitoring in the affected areas of the infrastructure to ensure normal behaviour is restored.

5.2 Results and Discussion

With typical operational data flowing in our simulated experiment, we introduced a flooding incident from H6 as seen in Figure 4. This was performed using *iperf* in UDP mode for a prolonged 60 second burst at 5x operational traffic levels to all hosts on the network: H_1 to H_5. Our experiment parameters were set with *MAX_NEW_ROUTES _THRESHOLD=2*. As the UDP flows began, the controller added new routes from H_6 to H_2, H_4 and H_5 (routes to H_1 and H_3 are already present). This took place within a five second timer event. After the flood had initiated, the next time the timer event was triggered, a Flood Characteristic Alert was successfully thrown to the network monitoring at the controller.

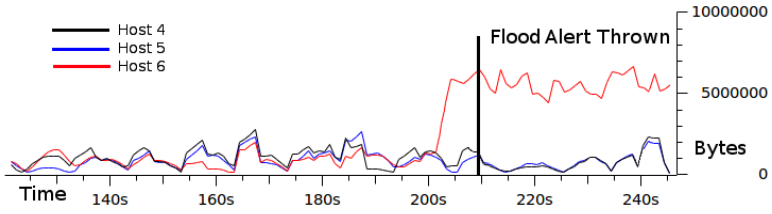

Fig. 4. Results of flooding and detection time plotted against normal operational traffic

Considering the lengthy outages which have occurred through flooding events in the past, we believe that this prototype experiment and our architecture shows strong potential for better securing and increasing reliability of critical infrastructures in the future.

Other techniques to detect flooding incidents exist of course. Preventative approaches such as VLAN isolation or another means of blocking access between hosts could be implemented, e.g., firewalls. However, flooding is one challenge from many in the set we have identified. While other techniques exist, the implementation of these can involve manually distributed hard-coding of policies which can be overly restrictive and unresponsive. Our architecture places the network operators in greater direct, responsive control of their network since they have the ability to deploy distributed modules tailored to the scenarios they perceive unfolding. We argue, our approach allows for greater flexibility and security with faster results than typical monitoring provides. In our experience, many network operators rely upon Simple Network Monitoring Protocol (SNMP) based network monitoring applications which typically poll network devices once every 15 minutes. While there is always a trade-off amongst the frequency of monitoring, the traffic this monitoring produces and the desire not to interfere with operational traffic, we believe our architecture presents an optimal solution. If additional monitoring is temporarily desired, with more granular frequency or reviewing a wider set of parameters, this can be deployed and processed at distributed points throughout the infrastructure, on-demand. By exploiting the decoupled control plane approach we recognised a flooding event within a few seconds and presented an alert detailing the switch at the root cause. The operator then has detailed information and can act rapidly.

6 Conclusions and Future Work

In this paper, we presented the case for adopting the latest programmable network technologies, specifically SDN, for the control and management of CNI. We reviewed a series of incidents which affected Air Transportation systems and distilled a set of applicable challenges and their characteristics. We presented a novel, modular SDN-based architecture and discussed the results of our simulated challenge on a scaled network topology, accurately modelled on an EU radar surveillance network using representative traffic, typical of operational flows. Our results showed our approach is viable, and coupled with programmable

networking principles, has strong potential for use in critical infrastructure data networks to detect challenges and abnormal behaviour. Future work will add to our simulated environment, creating a scaled version of our complete modular architecture and we will look at how to deploy security and monitoring components over the network on-demand. We intend to explore anomaly detection algorithms which we can distribute and tailor to exploit the characteristics of the challenges we present in this paper as well as attributes of critical infrastructures.

References

1. Dewaele, G., Fukuda, K., Borgnat, P., Abry, P., Cho, K. "Extracting hidden anomalies using sketch and non gaussian multiresolution statistical detection procedures." Workshop on Large scale attack defense, pp. 145-152. ACM, 2007.
2. Eurocontrol website, "Rome Fiumicino Airport becomes the 10th A-CDM airport". Accessed: 27/04/2014 `https://www.eurocontrol.int/news/rome-fiumicino-cdm-implementation-gears-critical-mass-full-benefits`
3. FAA FTI Review Panel, "Report on November 19, 2009 Outage", 2010
4. Flavel, A., Roughan, M., Bean, N., Shaikh, A. "Where's Waldo? practical searches for stability in iBGP." ICNP pp. 308-317. IEEE, 2008.
5. IAA, "Report of the IAA into the ATM System Malfunction", Sep 2008
6. Khurshid, A., Zhou, W., Caesar, M., Godfrey, P. "Veriflow: Verifying network-wide invariants in real time." SIGCOMM pp. 467-472. ACM 2012
7. Kuniar, M., Pereni, P., Vasi, N., Canini, M., Kosti, D. "Automatic failure recovery for software-defined networks." HotSDN, pp. 159-160. ACM, 2013.
8. Los Angeles Times, "LAX outage is blamed on 1 computer", Aug 2007
9. Mehdi, A., Khalid, J., Khayam, A. "Revisiting traffic anomaly detection using SDN." Recent Advances in Intrusion Detection, Springer 2011.
10. Mirkovic, J., Reiher, P. "A taxonomy of DDoS attack and DDoS defense mechanisms." SIGCOMM pp 39-53. ACM 2004
11. Open Networking Foundation, "SDN: The New Norm for Networks", April 2012
12. Press of Atlantic City, "Fire at Hughes Technical Center caused $2.2M in damage", Accessed: 27/04/2014 `http://www.highbeam.com/doc/1P3-2726195211.html`
13. Rinaldi, S. M., Peerenboom, J. P., Kelly, T. K "Identifying, understanding, and analyzing critical infrastructure interdependencies." Control Systems, IEEE 2001
14. Shalimov, A., Zuikov, D., Zimarina, D., Pashkov, V., Smeliansky, R. "Advanced study of SDN/OpenFlow controllers." CEE-SECR p. 1. ACM, 2013.
15. Sherwood, R., Gibb, G., Yap, K., Appenzeller, G., Casado, M., McKeown, N., Parulkar, G. "Flowvisor: A network virtualization layer". OpenFlowSwitch 2009
16. Shin, S., Porras, P., Yegneswaran, V., Fong, M., Gu, G. "Fresco: Modular composable security services for software-defined networks." Internet Society NDSS 2013.
17. Shirali-Shahreza, S., Ganjali, Y. "FleXam: flexible sampling extension for monitoring and security applications in openflow." HotSDN, pp. 167-168. ACM, 2013.
18. White, K.J.S., Pezaros, D.P., Johnson, C.W., "Increasing Resilience of ATM Networks using Traffic Monitoring and Automated Anomaly Analysis", ATACCS 2012
19. White, K.J.S., Pezaros, D.P., Johnson, C.W., "Principles for Increased Resilience in Critical Networked Infrastructures", Publication Pending. ICRAT 2014
20. Zhang, Y. "An adaptive flow counting method for anomaly detection in SDN." Emerging Networking Experiments and Technologies, pp. 25-30. ACM, 2013.