<http://www.diva-portal.org>

Postprint

This is the accepted version of a paper presented at *ICICS 2015 - The 10th International Conference on Information, Communications and Signal Processing, December 2-4 2015, Singapore..*

Citation for the original published paper:

Vestin, J., Kassler, A., Akerberg, J. (2015)

Resilient Software Defined Networking for Industrial Control Networks.

In: *2015 10th International Conference on Information, Communications and Signal Processing (ICICS)* IEEE conference proceedings

<https://doi.org/10.1109/ICICS.2015.7459981>

N.B. When citing this work, cite the original published paper.

Permanent link to this version:

<http://urn.kb.se/resolve?urn=urn:nbn:se:kau:diva-38748>

Resilient Software Defined Networking for Industrial Control Networks

Jonathan Vestin

Dept. of Maths and Computer Science
Karlstad University
65188 Karlstad, Sweden
Email: jonathan.vestin@kau.se

Andreas Kassler

Dept. of Maths and Computer Science
Karlstad University
65188 Karlstad, Sweden
Email: andreas.kassler@kau.se

Johan Åkerberg

ABB Corporate Research
Västerås, Sweden
Email: johan.akerberg@se.abb.com

Abstract—Software Defined Networking (SDN) is currently a hot topic in the area of Datacenter Networking or Enterprise Networks as it has the promise to radically simplify network management and operation. However, it has not been considered so far as a promising candidate for Industrial Control Networks mainly because of the deterministic performance requirements and the dedicated design of those networks to fulfil strict performance guarantees. In this paper, we propose a resilient SDN based architecture for Industrial Control Networks and show that by combining several SDN based fast failover technologies using per-link Bidirectional Forwarding Detection (BFD), pre-configured primary and backup paths and flexible packet duplication orchestrated by an SDN controller, we can reduce significantly the control latency and provide more stringent performance guarantees even under lossy and failing links.

I. INTRODUCTION

Software Defined Networking (SDN) has gained importance in the networking field because it allows a flexible de-coupling of control and data plane. In SDN, network elements are responsible for the pure forwarding of data packets based on forwarding rules which are computed and installed by a centralised controller using the OpenFlow protocol [1]. As such controller can be programmed in any high level language, this speeds up network innovation because it allows to deploy new services very fast. The controller may use and implement a large set of functions such as network monitoring, network re-configuration, etc.. This enables the controller to monitor the network for performance or faults and based on given policies reprogram the forwarding behaviour of the network on the fly. For example, the controller can detect link breaks based on e.g. packet loss statistics and compute alternative paths for the affected flows in order to restore end-to-end connectivity.

In Industrial Control Networks, the main requirement is timely data delivery with the individual deadlines of the data flows even in case of a single error. Therefore, one of the main challenges is to ensure the availability of the processes under control even in case of hardware failures and packet losses due to inherent interference. Industrial process control applications have extremely high requirements on availability and synchronized control, which poses strict requirements on latency, jitter and communication reliability [2]. Therefore, industrial control systems need to be designed in such a way that the process can continue even in case of a single packet loss, and be as resilient as possible in order to allow the process to be in operation 24/7 with short and planned

maintenance periods, typically once a year. As a consequence, those requirements make the design of Industrial Control Networks typically very inflexible and very expensive as they are based on many redundant proprietary complex network elements in order to cope with failures and latency constraints.

In this paper, we propose an architecture for Industrial Control Networks based on SDN and integrate several resiliency methods. The Industrial Controller has an interface to the SDN controller to configure the wired part of the Industrial Control Network according to the performance guarantees needed by the industrial application. By using flexible packet duplication using SDN forwarding rules together with a fast failover strategy, we can significantly reduce the latency and loss of sensor/actuator communication in the wired network part. To cope for link failures in the wired network, the SDN Controller precomputes a primary and one or multiple backup paths that are pre-installed at the forwarding nodes using fast failover group tables. Bidirectional Forwarding Detection (BFD) [3] protocol is used on each link to rapidly detect link failures by exchanging a rapid stream of control messages which locally triggers the switch to use the backup path once the primary path is considered down without involving the controller. An evaluation of our approach using the CORE network emulator has shown that our methods can significantly reduce the latency and packet loss for industrial control applications.

The remainder of this paper is structured as follows. Section II shows our architecture of SDN based Industrial Control Networks. Section III details the flexible forwarding and failover mechanisms that the SDN controller uses based on interaction with the Industrial Controller. Section IV evaluates the impact of different configuration parameters such as BFD monitoring interval and number of duplicates on the end-to-end performance. Section V concludes the paper.

II. SDN BASED INDUSTRIAL CONTROL NETWORK ARCHITECTURE

Industrial Control Networks typically consist of a wired and a wireless network. The wireless network typically consists of multiple wireless sensor nodes, actuator nodes, relay nodes and gateway nodes that connect the wireless (multihop) network to the wired Industrial Control Network. The sensor nodes retrieve a value from one or more sensors (e.g temperature sensor) and relay the information through the wireless (multi

hop) network towards the one or many Industrial Controller in the wired network. The actuator nodes wait for a signal from the Industrial Controller through the wired network over the wireless network and perform some action (e.g. activating a motor). In order to connect the (wireless) sensor and actuator nodes to the wired network, typically one or multiple gateway nodes are used. In such networks, it is crucial to provide dedicated performance guarantee between the sensor and actuator nodes in terms of e.g. latency and packet loss.

The most common strategy is to have redundant hardware in order to cope with single hardware failures, and redundant communication paths in order to improve the packet loss ratio due to interference and communication path failures. In addition, the concept of (m, k) -firm deadlines [4] is used in order to meet the stringent availability requirements and safety concerns. The (m, k) -firm deadline concept allows m consecutive deadline misses (including packet losses), but when k consecutive errors has occurred the process needs to be stopped for safety reasons. As a consequence, Industrial Control Networks have been designed for a given performance guarantee, which makes the network very costly and inflexible.

In this paper we focus on the wired network and assume that the wireless part is configured using e.g. TDMA type protocols to provide the required performance guarantees. The gateways then encapsulate/decapsulate the sensor/actuator data into packets that are routable over a SDN enabled network (e.g. using VXLAN or similar approaches). The encapsulation is done in such a way to be able to identify a given sensor/actuator stream of packets and treat each sensor/actuator flow differently in terms of resiliency required. That allows an efficient co-existence of different sensor-actuator pairs having different requirements in terms of packet loss and deadlines. We now focus on the resiliency mechanisms that SDN can provide in the wired part of the network and combine the benefits of network monitoring and greatly simplified and flexible network reconfiguration enabled by SDN.

In our architecture proposal, the SDN controller is responsible for configuring the wired network part of the Industrial Control Network. In particular, the SDN Controller exposes a Northbound API which allows the Industrial Controller to:

- Establish a primary and a configurable number of backup paths between gateway nodes and (multiple) Industrial Controller(s). The number of backup paths can be customised on a per sensor/actuator communication flow.
- Specify flexible packet duplication rules in order to increase the resiliency against packet loss in the wired network part. Again this should be configurable on a per sensor/actuator pair.
- Specify fast failover strategy parameters in order to cope with link failures in the wired network path.

Based on those requests, the SDN controller configures the wired network according to the application specific requirements of the Industrial Controller. In particular, it derives proper fast failover group table configuration using OpenFlow [1] and dedicated packet duplication rules that the controller installs at the given set of gateways, which we detail in the next section.

III. SDN BASED RESILIENCY FOR INDUSTRIAL CONTROL NETWORKS

Industrial Control Networks may run several time-critical applications that have such strict communication-delay constraints which make it difficult to implement retransmissions to cope with packet loss. In such applications, information needs to be exchanged between the sensor and the actuator in quasi-real time, within given hard delay bounds. The sensors typically stream periodic data to the Industrial Controller. If a packet is lost and retransmitted, the retransmitted packet may arrive at the controller anyhow too late to be useful. When run over IP networks, IP layer multicast can be used to disseminate the data to several controllers to increase redundancy. This calls for using UDP as a transport mechanism which makes it hard to provide reliability of unidirectional UDP traffic.

To this extent, the PRP protocol (Parallel Redundancy Protocol) [5] was designed for increased redundancy inside a LAN when there are no routers. It requires disjoint bridged networks and the sender replicates packets over the two networks using specifically tagged MAC frames containing sequence numbers that the receiver uses to discard duplicates. While PRP works well in controlled environments, where all elements are duplicated, a larger network requires routers to interconnect the LAN segments, rendering PRP useless. PRP requires the devices to have duplicate MAC addresses which complicates network design further. As PRP duplicates all packets, it may impose a high overhead in a routed network.

In this paper, we develop a solution for flexible replication of unidirectional unicast packets using SDN. We assume that the wired network has two or more parallel (ideally disjoint) paths from each gateway to the Industrial Controller. As current networks cannot control the path that packets take, the use of SDN simplifies the process significantly. As instructed by the Industrial Controller through the Northbound API, the SDN controller will calculate from each gateway towards each Industrial Controller a primary path as well as one or several backup paths for a given sensor/actuator traffic flow. Depending on the loss sensitivity of the industrial application, one or more backup paths will be calculated between a given AP that the sensor flow passes through towards one or more Industrial Controllers, which is subject to the specific industrial application to specify. Then, flow rules will be installed at the gateways that allow a flexible packet replication to the needs of the industrial application (i.e. an application that requires a very robust delivery service will install more duplicates than an application that requires less robustness). By using flexible flow rules, different sensor/actuator flows can coexist in the industrial network having different number of packet replications implementing different resiliency levels.

Our approach does not mandate disjoint networks although that would be beneficial. Any network redundancy can be exploited by the path calculation strategy at the SDN controller, which is outside the scope of this paper. Clearly, the availability of link or node disjoint paths would be beneficial in the network. It is also worth mentioning that it is not required to replicate all flows. Again, this is application specific and with our architecture we support the coexistence of multiple flows

that are protected in a different way by selective replication as dictated by the Industrial Controller.

If an intermediate wired link fails, it is however beneficial to detect this situation in a rapid way and reconfigure the network to restore connectivity as fast as possible. While an SDN based approach can be used to reconfigure the network, a reactive way may take a significant amount of time to recalculate the paths, as it needs to involve the SDN controller adding a round trip towards the controller to the path restoration time. This additional time might be prohibitive as packets may arrive during that restoration time that will be lost until end-to-end connectivity is restored. As a consequence, in line with [6] we propose to calculate for each link a backup link towards the destination that can be used to reroute packets locally in case link failure is detected. Again, this will be instrumented by the Industrial Controller through the Northbound interface to the SDN controller. In this paper, we assume that link break detection is performed locally at each router and a recalculated backup link towards the destination should be available immediately for fast-failover. Then the controller may recalculate end-to-end recovery and update the paths according to the requirements specified by the Industrial Controller.

As robustness and consistent latency is very important in Industrial Control Networks, a protocol that allows rapid detection of broken links is essential to the operation of a resilient network. However, fast ethernet was not designed for such robustness because it sends periodic heartbeats for link break detection in the order of roughly 20 ms [7]. If no response to such heartbeat message is received within say 50 to 150 ms, the link is declared down. For industrial control applications, a much faster reaction may be desirable. One approach is to use BFD [3], which implements a control and echo message mechanism which detects the up/down state of a network link. This works by establishing a connection using a three-way handshake, and periodically sending control messages over the link. The recipient of these control messages immediately replies with a echo message containing the link status. If no echo message is received within a certain time, the link is considered down. A typical timeout value is $4 * T_i$, where T_i is the control message interval, which is to prevent false positives in the link failure detection. BFD also adds a 0 to 25% jitter to its messages, to spread the impact on other network systems. By reducing the BFD control message, a rapid detection of link breaks is possible [6].

In OpenFlow 1.1 group tables were introduced, which allow forwarding of packets to one or more ports within a group. The Fast Failover group table forwards packets on the first output port. If a link failure is detected, the forwarding is immediately moved to the next output port in the group table. BFD can be used to facilitate the link state monitoring and notify the group table when to perform fast failover. We use the integration of BFD and fast failover group table from [6] to significantly reduce the time between link failure detection and using a pre-configured backup path. In this approach, we use BFD on a per link basis to trigger a fast router-initiated local path recovery based on preconfigured forwarding rules. As a last step, the controller restores end-to-end connectivity. This

requires that each router gets a preprogrammed backup path in terms of Fast Failover rules as instructed by the Industrial Controller depending on the applications resiliency requirements. In contrast to [6], we propose to allow a tunable control message interval, which we make configurable by the SDN controller. This allows to customise this important resiliency mechanism by the Industrial Controller to the characteristics of the Industrial Control Network.

IV. EVALUATION

In this section, we present an evaluation of the SDN based resiliency methods combining fast failover group tables with packet duplication. We focus in the evaluation on the wired part of the network. We used the CORE [8] network emulator that we extended to support OpenvSwitch and OpenFlow. Fig. 1 shows an overview of the network we used for the evaluation. The actuator receives a signal from the Industrial Controller, controlling a specific part of the industrial plant, through the wired and wireless network using multiple gateways. The sensors stream sensor readings through the wireless (multi hop) network through multiple gateways to the Industrial Controller. We evaluate latency and packet loss between the sensor and actuator via the Industrial Controller. As we focus on mechanisms in the wired network, we simplify the wireless part by assuming just a single hop but multiple gateway nodes.

Fig. 1: Evaluation Topology

We use two gateways and each switch has a primary and backup path configured in such a way that a switch can only forward a packet to another switch which has fewer hops to the Industrial Controller in order to avoid loops. Thus, the primary path for $K1$ would be $K1 \rightarrow C1 \rightarrow B1 \rightarrow A1 \rightarrow IC$. There is also a redundant Industrial Controller connected to the network, in case the active one fails. For the wired network, we have configured 1 ms latency for each link, while the latency for the wireless portion is 5 ms. We do acknowledge the fact that this is a very simple assumption but the control of the wireless part is outside the scope of this work. However, some links have special latency parameters: $A1 \rightarrow IC$: 1 ms; $A2 \rightarrow IC$: 5 ms and $A3 \rightarrow IC$: 10 ms. The wireless network is simulated using a single node, which sends UDP packets every 20 ms to the one or multiple gateway nodes. The UDP

packets are 8 bytes in size (not including headers) and contain the packet type, a sequence number and 32 bits of data from the sensor. The gateway nodes perform packet duplication (if instructed by the SDN controller), and forwards these packets towards the Industrial Controller using pre-calculated disjoint paths. The Industrial Controller receives the packets and, based on the sequence number discards duplicated packets.

The bandwidth of each link is configured to 1 Gbps with 1% loss in the wireless network and 0% in the wired network. However, the links $C1 \rightarrow B1$, $C2 \rightarrow B2$ and $C3 \rightarrow B3$ use the Gilbert-Elliot loss model with the parameters $p = 1$, $r = 19$, $1 - h = 81$ and $1 - k = 1$ to introduce bursty losses in the wired network. In order to simulate link failures, the link $A1 \rightarrow B1$ is brought down after 60 s and the link $A2 \rightarrow B2$ is brought down after 120 s (the whole test is 180 s long). We run different tests in order to evaluate the impact of the BFD monitoring interval and the packet duplications on the packet loss rate and latency distribution.

Fig. 2 shows the ECDF of the RTT (Round-Trip Time) for the packets sent from the Sensor towards the Industrial Controller and back to the actuator using different number of Gateway Nodes and packet duplicates. We compare the following cases: a single gateway sending 1 copy (1 GW, 1 Copy), a single gateway sending duplicate packets (1 GW, 2 Copy), 1 gateway sending 3 copies of each incoming packet (1 GW, 3 Copy), 2 gateways each sending 2 copies each (2 GW, 2 Copy) and finally 2 gateways each sending 3 copies each (2 GW, 3 Copy). For 2 copy cases, GW1 sends a packet over C1 and a duplicate over C2 while for 3 copy cases, GW1 sends a packet over C1, and duplicates over C2 and C3, respectively. In the 2 GW case, we assume that the wireless sensor network has delivered 2 identical packets to the two different gateways. Intermediate wired network links run BFD to detect link losses and perform fast failover based on SDN without creating additional duplicates.

Fig. 2: 10 ms BFD control message interval

When using 1 GW 1 Copy and a 10 ms BFD interval, 11% of the packets have a RTT lower than 20 ms, and 84% of the packets have a RTT lower than 25 ms. When the gateway sends 2 copies, 21% of the packets have a RTT lower than 20 ms, while 90% of the packets have an RTT lower than 25 ms. For 3 copies, this increases to 26% vs. 95%, respectively. This is a

consequence that we disperse the copies over spatially disjoint paths so we get a higher probability that some packets arrive earlier. In the 2 GW case, sending 2 copies results in 23% of the packets to show a RTT lower than 20 ms (94% below 25 ms). For the 2 GW, 3 Copy case, we can see similar results. Having more copies and more gateways reduces the latency in the network. Note, that in our case the network is not fully loaded and the additional overhead to send the copies has no impact on the congestion as the amount of sensor data is small compared to the network capacity.

Fig. 3: 20 ms BFD control message interval

We increased the BFD control message interval to 20 ms and the results are depicted in Fig. 3. As we can see, when using 1 GW, 1 copy, the probability to have a RTT lower than 20 ms increases from 11% to 18% while the probability to have latency below 25 ms decreases from 84% to 80%. For the 2 Copies case, the probability to have latency lower than 20 ms increases to 28% (RTT lower than 25 ms decreases to 89%). For the 3 Copies case, 26% of the packets have a RTT lower than 20 ms and 92% of the packets have a RTT lower than 25 ms. As for using 2 gateways sending 2 copies, 22% of the packets have a RTT lower than 20 ms while 93% of the packets have a RTT lower than 25 ms. Using 2 gateways, each sending 3 copies, 20% of the packets have a RTT lower than 20 and 94% of the packets have a RTT lower than 25 ms.

Fig. 4: 50 ms BFD control message interval

Fig. 4 shows the same tests, but with the BFD control message interval increased to 50 ms. Results show a similar

trend. As we can see from those CDFs, using a BFD control message interval of 20 ms or 50 ms as opposed to 10 ms results in general in higher latency in particular without packet duplication. Without packet duplication, packets are forwarded using a single path, where the aggregate RTT increases to 20 ms after 60 s and 25 ms after 120 s. However, if packets are duplicated, there is a higher probability that some packets take the faster paths which brings down the latency.

Setup	PLR (10ms)	PLR (20ms)	PLR (50ms)
1 GW, 1 Copy	4.9%	5.6%	7.8%
1 GW, 2 Copy	2.3%	3.1%	5.0%
1 GW, 3 Copy	2.7%	3.4%	4.9%
2 GW, 2 Copy	0.2%	0.2%	0.2%
2 GW, 3 Copy	0.2%	0.6%	0.2%

TABLE I: Packet Loss for different BFD echo intervals

Table I shows the packet loss ratio for different schemes and BFD intervals. PLR (10ms) depicts the packet loss ration when the BFD echo interval is 10 ms, while in column PLR (20ms) we increase this interval to 20 ms, etc.. As we can see, having more copies reduces significantly the packet loss probability. In the 2 GW case, we can almost achieve zero loss probability despite our worse case assumptions on link loss models. This shows that our resilience methods are very effective in providing an almost error free delivery service. However, when the echo interval increases from 10 ms to 20 ms and 50 ms, the packet loss ratio increases. This is because when a link is shut down, it takes more time for the BFD to detect this situation and shift the packets to the backup path, causing packets to be dropped for some longer time. The 2 copy and 3 copy case have similar PLR as the loss occurs in the wireless portion of the network, before packet duplication is performed.

V. CONCLUSION

SDN is an interesting alternative to simplify the configuration, management and operation of future networks. To this extent, we have proposed an architecture that allows to apply the paradigm of SDN to Industrial Control Networks that traditionally have been engineered to provide customisable resiliency. In our architecture, the SDN controller is used for the wired part of the industrial network and provides a dedicated interface, which enables the Industrial Controller to configure the forwarding in the wired network in a flexible and resilient way. In order to provide resiliency against packet loss and link failures, the controller can apply two techniques. The first one is flexible packet duplication and the second one is based on fast failover. The SDN controller pre-configures a primary and (multiple) backup paths between the gateways and the Industrial Controllers. Each links use local BFD sessions to rapidly detect and react to link failures locally.

In relation to traditional Industrial Control Networks, we can conclude the following. There are several redundancy mechanisms which an SDN based architecture enables to apply in the context of Industrial Control Networks. We have shown that by applying forwarding rules, an SDN based architecture can provide a very flexible approach for packet duplication

and individual routing towards one (or multiple) Industrial Controllers. Depending on resiliency demands, the Industrial Controller can instruct the SDN controller to increase redundancy. It could do that even on a per sensor/actuator flow basis. This is in contrast to PRP, which has a very inflexible approach as it treats all sensor data the same way. An SDN based approach thus allows a very flexible network configuration and enables a dynamic change in resiliency, even during the normal operation by changing the flow rules. This is in contrast to traditional Industrial Control Networks that would require a shutdown of the network in order to reconfigure its operation. An SDN based architecture provides thus much more flexibility and simplifies network operation and management for Industrial Control Networks.

While an initial evaluation using CORE emulator shows that the SDN controller can actually configure the network to reduce latency and packet loss, many issues need to be resolved in future work. For example, how the primary and backup paths are calculated need to be studied in more detail along with the configuration of the wireless part of the network. Also, we intend to study how to prioritise different packets in a more flexible way according to the demands of the industrial application. Finally, an evaluation with a real testbed should provide new insights into reliability issues in such harsh environment.

ACKNOWLEDGMENTS

This work is supported by the Knowledge Foundation through the project READY.

REFERENCES

- [1] Nick McKeown et al. *OpenFlow: Enabling Innovation in Campus Networks*. Tech. rep. Mar. 2008.
- [2] J. Akerberg, M. Gidlund, and M. Bjorkman. "Future research challenges in wireless sensor and actuator networks targeting industrial automation". In: *Industrial Informatics (INDIN), 2011 9th IEEE International Conference on*. 2011, pp. 410–415. DOI: 10.1109/INDIN.2011.6034912.
- [3] D. Katz and D. Ward. *Bidirectional Forwarding Detection (BFD)*. RFC 5880 (Proposed Standard). Updated by RFC 7419. Internet Engineering Task Force, June 2010. URL: <http://www.ietf.org/rfc/rfc5880.txt>.
- [4] M. Hamdaoui and P. Ramanathan. "A dynamic priority assignment technique for streams with (m, k)-firm deadlines". In: *Computers, IEEE Transactions on* 44.12 (1995), pp. 1443–1451. ISSN: 0018-9340. DOI: 10.1109/12.477249.
- [5] M. Rentschler and H. Heine. "The Parallel Redundancy Protocol for industrial IP networks". In: *Industrial Technology (ICIT), 2013 IEEE International Conference on*. 2013, pp. 1404–1409. DOI: 10.1109/ICIT.2013.6505877.
- [6] N.L.M. van Adrichem, B.J. Van Asten, and F.A. Kuipers. "Fast Recovery in Software-Defined Networks". In: *Software Defined Networks (EWSN), 2014 Third European Workshop on*. 2014, pp. 61–66. DOI: 10.1109/EWSN.2014.13.
- [7] L. Wang, R.F. Chang, and C.S. Yik. "Apparatus for link failure detection on high availability ethernet backplane". In: *US Patent 7,260,066* (2007).
- [8] J. Ahrenholz. "Comparison of CORE network emulation platforms". In: *MILCOM 2010*. 2010, pp. 166–171. DOI: 10.1109/MILCOM.2010.5680218.