Resource-aware Virtualization for Industrial Networks

A Novel Architecture Combining Resource Management, Policy Control and Network Virtualization for Networks in Automation or Supervisory Control and Data Acquisition Networks

Hans-Peter Huth and Amine M. Houyou
Siemens AG, Corporate Research, Munich, Germany

Keywords: Network Virtualization, Quality-of-Service, Automation Networks, Industrial Ethernet, Network Management and Control, Multi-tenancy.

Abstract: Within the EU Project IoT@Work future Internet techniques such as network virtualization are investigated in order to provide communication services in automation systems. The physical network is modeled as an abstract pool of resources that is allocated through virtual slices to different automation applications. Each slice of the network resources fulfils the application communication needs in terms of security, QoS, and reliability. The proposed architecture is targeted at constrained networks, where a centralized control and management point could be accessed through an API to create and implement virtual networks which could be mapped to a real physical infrastructure. This approach would impact not only the task of engineering by separating application programming from configuring networks. It also hides the network dynamics and self-management connecting ever more dynamic, adaptive and smart automation systems.

1 INTRODUCTION

Networks play a major role in all distributed applications and any network has to support many, sometimes very different, data flows which are potentially planned, implemented and operated by different parties. Consequently configuration of networks to meet those requirements is a crucial task. In telecommunication networks and to a lesser degree also in enterprise networks the solution to achieve the desired quality of service (QoS) is to construct virtual networks dedicated and optimized for the respective applications which try to optimize the use of the underlying physical network. There exists a broad range of possible network virtualization solutions adapted to the respective domains (Duan et al., 2012)

In the last years, networks controlling machines in industry automation, buildings or in Supervisory Control and Data Acquisition (SCADA) systems are using more and more technologies and standards which are built upon Ethernet and Internet technologies. However, those networks face very specific requirements which cannot easily be met by applying technologies from the telecommunication or enterprise networks. Still, there is an ever increasing need for interoperability driven by higher integration between clouds, enterprise networks, on the one hand, and the industrial networked systems on the other. Several trends in these areas demand for new networking concepts. First, the upcoming Internet-of-Things will vastly increase the number of devices and applications managed and to be supported in the network and will demand for more agility, see also Houyou et al., 2012. Second, the market demands for an ever-increasing flexibility to support new business models and to be able to quickly adopt new technologies. In the industrial domain, however, it is crucial to provide a reliable and deterministic operation. This is the major reason why operators of industrial networks are reluctant to change a running system.

This work introduces a system architecture which adapts the network virtualization and resource control ideas to the industrial domain to achieve a greater agility, lower management and control costs while still fulfilling industrial requierements. Specifically we use the industry automation as a major focus as it has both high complexity and tight requierments so it is a good benchmark for the novel solution.

Huth H. and M. Houyou A..
Resource-aware Virtualization for Industrial Networks - A Novel Architecture Combining Resource Management, Policy Control and Network Virtualization for Networks in Automation or Supervisory Control and Data Acquisition Networks.
DOI: 10.5220/0004508900440050
In *Proceedings of the 4th International Conference on Data Communication Networking, 10th International Conference on e-Business and 4th International Conference on Optical Communication Systems* (DCNET-2013), pages 44-50
ISBN: 978-989-8565-72-3
Copyright © 2013 SCITEPRESS (Science and Technology Publications, Lda.)

Industrial automation and control are characterized by a great variety of communication standard. There is still a large installation of non-IP based so-called fieldbus systems, but in this study we address IP-based technologies only as this is commonly considered to be the future in automation (see Jasperneite et al., 2009).

Requierements for automation networks differ significantly from their counterparts in enterprise and telecommunication. Jasperneite 2009 lists, e.g., low configuration costs, IEEE-802.3 compatibility, hard real-time, and reliability requierments far beyond multimedia for automation networks. Another non-technical requierement has its origin in the long life cycles and large base of installed -and often very expensive- devices in industrial domains. Thus any new concept must provide a clear migration path that enables gradual modernisation by re-using as much as possible from existing installations. Shielding control applications from other traffic in the LAN is requiered for reliable operation (Martin, 2011). Devices in the embedded domain typically also differ from enterprise or telecommunication equipment. These devices are in general much smaller in terms of computing power and number of ports. Also, - as opposed to application-based programming- many real-time related functions are supported by ASIC implementations and specialized operating systems. However, most of these devices already have powerful management features by means of standard SNMP and propietary extensions. There is also a high cost pressure on both network and end devices in automation. Thus all solutions must be “lean,” i.e. they should not add additional costs by requiering powerful CPUs or larger on-device memory.

Furthermore, both network and applications typically are subject of a planning step. After this step there is the commissioning of the configurations determined in the planning phase, followed by the operation phase. One drawback of this approach is that any change in the applications or in the network may necessitate revision of the planning. If this is the case, reconfigurations in many places might be required. Additionally, many applications are tightly bound to a select physical topology and specific devices. This tight coupling of applications with the physical network makes again any change a challenge.

The solution presented here decouples applications from the physical network. This is done by providing an abstract view that is independent of concrete technologies and that is also independent of the physical network topology. Furthermore, different applications can be seperated and “shielded” from each other by means of on-demand virtualization of the network and its resources. This allows for independent changes in applications or in the physical network. By adherring to this approach one also isolates changes in a way that is not needed to re-plan the complete automation applications if only small changes have occured. Forming virtual networks for applications also allows to integrate different technologies. Our apporach has thus the additional avantage of supporting heterogeinity and multi-tenancy in a natural way.

Unlike telecommunication networks, our approach cannot be realised by using overlays such as VPNs or MPLS. The reason for this is that the underlying network shall remain simple and well suited for small, embedded devices. Also, the per-packet overhead of tunneling technologies (overlays) makes it impossible to meet ultra-fast forwarding and switch times required in industrial-Etherent standards. Even lean concepts like MPLS may not be usable, since they requiere reading larger portions of a packet before a forwarding decision can be met. This procedure prohibits ultra-fast cut-through forwarding found in many industrial switches (see Prytz, 2008).

Since overlays cannot be used for virtualization in automation networks, other means are needed. Our approach attempts to re-use layer-2 features such as, e.g., VLANs (see Chowdhury and Boutaba , 2010) or Shortest-Path Bridging (see D. Allan et al., 2010). We also consider the direct manipulation of forwarding/routing tables in the the spirit ofOpen Flow (see, e.g., Curtis et al., 2011). Unlike Open Flow however, our concept allows to implement a work split between an intelligent and central live management tool and intelligent devices, which then can operate at full speed and with some autonomity.

Because our architecture essentially partions a network including resources into new virtual networks, which -from an applications point of view- behave more or less like physical networks, we use the term 'slice' for a virtual network in our architecture (see Chowdhury and Boutaba, 2010).

2 RELATED WORK

Network virtualization is not new. Well known technologies and standards include IEEE 802.1ak VLAN (IEEE Std 802.1Q and amendments) and Multiprotocol Label Switching (Rosen et al., 2001). These technologies allow marking and identifying packets throughout the network and also

manipulating the way packets are treated (i.e. routing, QoS, security policies). This is achieved in all underlying network elements such as switches or routers. IP tunnelling or encrypted SSH tunnels is a second form of network virtualization, also called 'Virtual Private Networks' (VPN). VPNs create an overlay between peers and cannot easily be manipulated by the underlying infrastructure.

Current network-virtualization approaches address the following problems by manipulating the route of packets between end points:

- Logical separation of distributed applications or end points over the same shared physical infrastructure (path isolation). The result of such a logical separation is frequently referred to as 'substrate'.
- Allow additional security features (encryption, access control, and authentication) and limiting connectivity to parts of the network.
- Less common, but also possible, is to tailor different QoS dimensions (e.g. bandwidth, delay guarantees, reliability needs, security, etc.) for each virtual network. This is done by hiding the substrate choices or the methods involved in establishing the virtual network. Therefore, applications are unaware of the underlying resource management and resource allocation. Both are managed on a per-virtual-network basis.

Additionally, one needs means to (a) calculate optimal paths through the network (traffic engineering) and (b) means to commission rules to network elements. The latter is necessary for configuring the virtual network. The state of the art is to rely on costly and complex tools that are inherently bound to a specific technology. Step (a), traffic engineering, is typically done off-line and then commissioned once before network boot. There are rare exceptions, for example the MPLS- and IP-routing-based approach described by Hoogendoorn 2003. Similar to Hoogendoorn work, this paper also introduces a domain controller that is responsible for part of the network. This controller also implements online traffic engineering and automatic configuration of network elements.

Together with a better management of resources in the network and a classification of communication needs of “factory” applications, we aim at making the network management happen in an autonomous way. This is done in such a way that requests for resources are dealt with in a central entity which also may established and/or extended virtual networks as needed. Also, changes (for optimization purposes, or dealing with limited failures) in the network are hidden from both the operator and the application as much as possible. By so doing one limits the runtime management overhead to dealing with critical situations (e.g., large-scale hardware failures or system breakdowns).

However, today's solutions are pre-configured by means of network management based on pre-planned rules, and resource management or QoS aspects are typically not or only rudimentary handled. The granularity of a solution is typically on a domain or application-group level. Examples include multi-tenancy networks enabling many companies to share a common infrastructure or to separate various applications such as voice-over-IP, network management and automation. Our approach shall support a finer-granularity mapping of services to virtual networks. While this is of course feasible with today's approaches, the incurred management costs prohibit implementation and management of a (potentially) large number of virtual networks. In fact, today's networks do not have any interface to applications for creating, changing, or even monitoring properties of a virtual network. The proposal of combining several Ethernet extensions and protocols with virtualization has been addressed in previous work. However, the proposed combination by Finn 2012 does not address the network-design issue and the interface between applications and the network.

Rather than using a zoo of specialized protocols another approach is to directly configure the behaviour of network elements. Notice that the this strategy is also pursued by the OpenFlow community (The GENI Project Office, 2008). OpenFlow however imposes a high signalling overhead, as it typically attempts to control the network on a per-flow level. Also, OpenFlow requires new network devices (Curtis et al., 2011). Another weak point of OpenFlow is error handling: rearranging or recreating paths and states may take a significant amount of time due to a high signaling load. To meet industrial reliability demands, very fast and locally implemented failure recovery is a must.

The interface to a virtual network for applications is typically mapped to a physical or a virtual network interface (see Fischer et al., 2013). There are basically two strategies for achieving this mapping:

- An end device is attached to an edge device (i.e. a switch) and the virtualization begins/ends at the respective edge port (i.e. port-based VLAN). In this case, all applications on the end device will share the same virtual network and are not be able to detect it.
- A virtual network ends in the end device and a virtual interface provides network access. In this

case, applications can use a specific virtual network by simply using the respective virtual interface.

Building on these existing protocols and technologies, our communication service adds means for creating and tearing down a virtual network on demand, and for managing resources “behind the scenes.”

In the state-of-the-art solutions presented above, the network does not -or only in a very restricted way- communicate with applications. Thus, applications that need to react to network issues such as broken links or bandwidth problems, need to implement either their own end-to-end monitoring system or they need to use the minimal error messages provided by the socket layer via ICMP. Our approach on the one hand shall provide a richer interface for network state information. On the other hand it shall hide and repair many errors by means of resilience mechanisms.

Network virtualization is in principle independent from quality-of-service or other traffic policies, but in telecommunication a combination of both already in use. One example is the use of MPLS to create virtual networks and the use RSVP-TE to apply QoS policies to that path (Rodriguez-Perez and Gonzalez-Sanchez, 2007). This approach may employ different granularities, i.e. it may operate on aggregates or per-flow.

There are QoS concepts on different OSI layers that are in principle independent of network virtualization. For example IntServ (Wroclawski, 1997) is based on reservations for application-layer flows. A similar idea is used on layer-2 by the Audio/Video Bridging (AVB) standards promoted by IEEE 802.1 (Imtiaz et al., 2009). These approaches, however, rely on a routing or path finding mechanism that are hard to manipulate in order to allow full-blown traffic engineering similar to MPLS (i.e. assigning different parallel traffic to different paths).

It should be noted that today’s concepts of network virtualization and network-resource control are also driven by cloud-computing and data-center applications. In these case, creation of virtual networks on-demand is an essential functionality, the so-called 'Network-as-a-Service' (Costa et al., 2012).

3 SOLUTION ARCHITECTURE

3.1 Overall System Architecture

The solution presented here proposes, at its core, a network-abstraction layer, the 'slice layer' (see fig. 1).

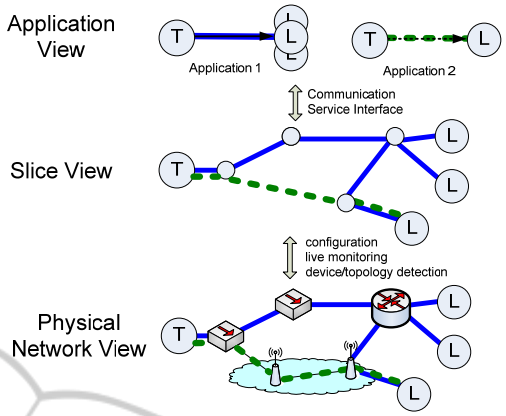

Figure 1: Layers of Network Abstraction.

This slice layer provides a graph view of the network: applications can only see edge nodes and intelligent traffic-engineering algorithms can operate on the graph without knowing details of the underlying substrate implementation. Important architectural components are clean interfaces meaning they must provide clearly defined responsibilities and enforce a "separation of concerns" (Laplante, 2007). Applications (including planning and management tools) use the 'Communication Service Interface' (CSI) for setting up and using the network. The slice view is provided and controlled from a central component, the slice manager (SMGR). The SMGR is responsible for a network, or a part of a network. If a network virtualization (= slice) must be constructed, the SMGR receives the necessary specifications from the application via the CSI. The SMGR then calculates an optimal routing sub-graph and finally commissions the required rules to the physical network using a 'driver layer'. The driver layer has knowledge of the concrete methods for accessing the devices and this layer also provides device abstraction for the slice view in the SMGR. The driver layer forms the base for including a multitude of different standards or vendor-specific device interfaces. It is an important component for making the slice layer agnostic to the underlying device and protocol heterogeneity. Unlike approaches like OpenFlow or MPLS based networks, this layered approach does not mandate for specific interfaces. The slice manager can handle all of the interfaces simultaneously by means of driver "plug-ins". Because of that, the slice manager constitutes an upgrade-friendly solution.

Devices, be it end devices or network elements, must be able to perform routing or forwarding according to the policies defined by the SMGR. In our

architecture, we bundle the needed functionality plus signaling interface towards the SMGR in a logical function called 'Slice Enforcement Point' (SEP, see fig. 2). The SEP can be implemented on that device or it uses other means (e.g. SNMP) to remotely perform its tasks. End devices that are not under control of a SEP -hereafter called 'legacy devices'- can be attached to an adjacent network element with an SEP ('edge SEP' in fig. 2). This edge SEP controls the corresponding network interfaces and enforces the needed policies for the legacy devices. This solution is similar to a port-based VLAN, which indicates a migration path from today's Ethernet technology toward our slice solution.

Devices that are specifically dedicated to support the new approach will have their own local SEP ('integrated SEP', fig. 2) so that slices can origin in that device.

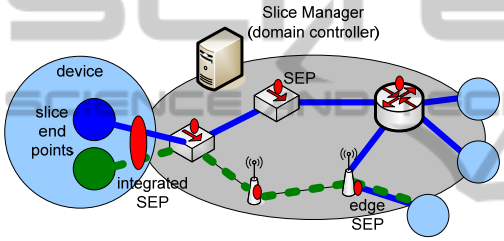

Figure 2: Architecture Birds View.

3.2 The Slice Manager

The slice manager controls all devices it is responsible for and it provides the interface for managing slices. Fig. 3 shows the internal functional architecture of the slice manager.

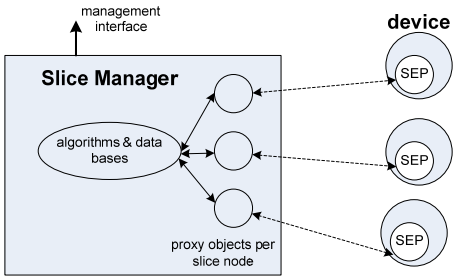

Figure 3: Simplified Slice Manager Architecture.

The slice manager needs exposes both a management interface and an interface for accessing devices in the network. The management interface allows controlling (i.e. establishing, tearing down and changing) slices without interventions of applications. The management interface can also be accessed by planning tools or by an operator. The interface towards the physical devices follows an object-oriented paradigm with one proxy object per device managed by this SMGR instance. These proxy objects also provide a standardized abstract view of a device and they contain drivers for accessing the device. For example, an industrial-Ethernet device and a legacy-Ethernet device have different proxy implementations and the algorithms in the SMGR only see different device properties but access to the devices (viz. the interface) is of the same type for both.

In order to manage and optimize slices, the SMGR must not only know all devices and their capabilities, but also the topology of the network. One way to accomplish this is to collect all neighborhood information from the devices and infer the network graph from the information collected. As SNMP and LLDP are already frequently used in industrial devices, this info is usually present even in today's devices.

Notice that fig. 3 does not show an application interface. We propose to re-use the device interface (which uses SEP signaling) as an application signaling channel for several reasons: (a) it is easy for applications to discover a local SEP, so there is no need for another service discovery. (b) Access control becomes easier as it already can be performed in the SEP. (c) No need for an additional asynchronous and multi-client interface, a fact that simplifies implementation of the SMGR.

3.3 The Slice Enforcement Point

As explained before, the SEP is a functionality bound to a device. It is responsible for signaling towards the SMGR or the applications, and it controls the network interfaces of that device. For example, the SEP must be able to manipulate forwarding or routing tables and to set QoS rules. The actual implementation can re-use existing control interface such as SNMP (Case, 2002) or IETF FORCES (Yang et al., 2004). However, we believe that a small dedicated agent forming the SEP is beneficial because it can be tailored for that purpose and it may add local intelligence which, e.g., enables quick recovery in failure cases without contacting the SMGR. Notice that due to the object-oriented architecture of the SMGR, a mix of different SEP implementations is of course possible.

3.4 The Applications Interface to Communication Services

In order to use slices, applications need a

communication interface (User Plane) and signaling means (Control Plane) for attaching to a slice and for specifying properties such as e.g. bandwidth constraints or QoS. For the user plane we propose the use a virtual layer-2 interface, similar to solutions found in server-virtualization environments. The signaling can use the next SEP as entry point, which is either located on a device or on an adjacent edge device (switch or router).

Specification of a slice includes at least a specification of the required QoS, but in order to enable online traffic engineering, a specification of the traffic is also beneficial. Notice we propose to use slices for aggregate flows of an application and not for single flows, although the latter is conceptually possible.

Beside pure QoS, industrial applications may have tight reliability requirements. Further more, some of them will have a high importance (i.e. the safety application). Thus we propose to add a notion of resilience and importance to the specification of traffic requirements. A traffic-engineering algorithm can use this information to, e.g., establish redundant paths or to resolve resource shortages.

3.5 Use Cases

The following sample use cases illustrate how the slice system can be used.

First, consider a data acquisition application accessing many sensors and a server collecting the sensor data. In a planning step, a traffic matrix is calculated and a slice is defined. The slice definition can be forwarded to the slice manager. The slice manager then calculates an optimal path for the slice and commissions it to the network devices. After that, the end devices (sensors and server) simply attach to the predefined slice. In the 'attach' process, a virtual network interface is created by the device SEP and the interface is bound to the slice in question.

A second example is an automation application that uses a PLC for controlling actuators and reading sensors. The application requires “hard” real-time communication and is programmed by a planning tool. This tool can estimate and generate the traffic matrix and also the slice specification. The latter is then pushed into the slice manager. Unlike today, the planning tool does not need full knowledge of the topology and capabilities of the network. The slice manager in turn - knowing the properties of the network - can construct an optimal slice, eventually re-using specific means from one of the many Industrial Ethernet standards (Jasperneite et al., 2009).

While the first two examples assume pre-planning, dynamic slice set-up in the running system can is also feasible. For instance, consider that a remote support needs to access a robot for maintenance. In this case the factory operator may install means for granting access for the remote service over the firewall. In order to facilitate this, the operator installs a slice on the fly. This slice might support best-effort communication and an upper bandwidth limit. Additionally, this slice only exposes the devices needed for the remote service.

These sample use cases also illustrate that the 'normal' use of the slice system is to construct semi-static or longer living slices for aggregates (i.e. flows belonging to one application). In many cases the slices will be instantiated by means of a configuration/management interface rather than application signaling.

3.6 Security Aspects

Slices or network virtualization in general is not a security concept; rather it is a complements security tools for constructing application 'sand boxes'. Furthermore, the proposed approach allows not only to dynamically grant or deny access, but it also supports 'on-demand' control of the resources that a certain slice consumes. By virtue of this control the slice manager prevents DoS attacks (see also the third use case above). We believe that this is an essential property for supporting multi-tenancy and applications sharing the same infrastructure.

4 CONCLUSIONS

The work presented specifies a system architecture for network virtualization in industrial networks. Concepts from Enterprise and Telecommunication have been mapped and adopted to fulfill industrial requirements. A domain controller, the Slice Manager, directly manipulates a potentially heterogeneous network while providing a simple abstract view to planning and management applications. Applications run in *"slice"* which is a virtual network with clear QoS guarantees and bandwidth policies.
Ongoing research focuses on three issues:

- Scalability: our current approach assumes one slice manager. In order to cover larger physical areas (signaling lag!) or larger numbers of devices, we investigate the use of multiple domain controllers, which, of course, need to co-ordinate their work.
- Optimal use of Industrial Ethernet features: current industrial standards provide many advanced resili-

ence features, sophisticated QoS beyond standard Ethernet, and many other features optimized for industrial requirements. Re-using these capabilities in an optimal manner is another challenge for future work.

- Traffic Engineering adapted to the slice system: considering the specific capabilities and requirements in this industrial domain, TE algorithms known from other areas must be analysed and adapted.

Further more, implementation issues and standardization have to be considered. The first topic is addressed by a Linux-based proof-of-concept, which forms a test bed for future enhancements. Concerning standardization, the authors believe that engaging IETF Forces or the OpenFlow community may be a way to proceed, but also Industrial Ethernet standardization must be addressed.

ACKNOWLEDGEMENTS

This work has been carried out in the European FP7 funded research project IoT@Work.

We would like to thank Jo Walewski for his useful review and helpful discussion.

REFERENCES

Houyou A., Huth H.-P., Kloukinas C., Trsek H., Rotondi D., *"Agile Manufacturing: General Challenges and an IoT@Work Perspective"*, 17th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2012); 09/2012

Duan Q, Yan Y., Vasilakos A., *"A Survey on Service-Oriented Network Virtualization Toward Convergence of Networking and Cloud Computing"*, IEEE Transactions on Network and Service Management, Vol. 9, No. 4, Dec. 2012

Jasperneite J., Imtiaz J., Schumacher M., Weber K., *"A Proposal for a Generic Real-Time Ethernet System"*, IEEE Transactions on Industrial Informatics, 06/2009

Hoogendoorn C., Schrodi K., Huber M., Winkler C., Charzinski C., *"Towards Carrier-Grade Next Generation Networks"*, Proc. ICCT 2003, Beijing, China, Apr. 2003

The GENI Project Office, *"The GENI System Overview"*, Document ID: GENI-SE-SY-SO-02.0, September 29, 2008, Cambridge, US, geni.net

Rosen E., Viswanathan A., Callon R., "*RFC 3031: Multiprotocol Label Switching Architecture"*, IETF, 2001

McKeown N., Anderson T., Balakrishnan H., Parulkar G., Peterson L., Rexford

Curtis, A., Mogul, J., Tourilhes J., Yalagandula P., Sharma P., Banerjee S., *"DevoFlow: Scaling Flow Management for High-Performance Networks"*, SIGCOMM 11, August 15-19, 2011, Toronto, Canada

Martin K., *"Network Infrastructure in Large, Integrated Control Systems"*, Cement Industry Technical Conference, 2011 IEEE-IAS/PCA 53rd

Costa P., Migliavacca M., Pietzuch P., Wolf A. 2012. *"NaaS: network-as-a-service in the cloud."* HotICE'12. USENIX Association, Berkeley, CA, USA

Case J., *"Introduction and Applicability Statements for Internet Standard Management Framework"*, RFC 3410, IETF, Dec. 2002

Yang L., Dantu R., Anderson T., Gopal R., *"Forwarding and Control Element Separation (ForCES) Framework*", RFC 3746, IETF, April 2004

Wroclawski J., *"The Use of RSVP with IETF Integrated Services"*, RFC 2210, IETF, Sept. 1997

Imtiaz J., Jasperneite J., Han L., "*A performance study of Ethernet Audio Video Bridging (AVB) for Industrial real-time communication*", Proceedings of 12th IEEE International Conference on Emerging Technologies and Factory Automation, Palma de Mallorca, Spain, 22-25 Sept. 2009

Chowdhury N., Boutaba R., "*A survey of network virtualization*", Computer Networks, 2010 – Elsevier, Volume 54, Issue 5, 8 April 2010, Pages 862–876

Allan D., Ashwood-Smith P., Bragg, N. Farkas J., Fedyk D. Ouellete M., Seaman M, Unbehagen P., "*Shortest path bridging: Efficient control of larger ethernet networks*," *Communications Magazine, IEEE* , vol.48, no.10, pp.128-135, October 2010.

Rodriguez-Perez F., Gonzalez-Sanchez J., "*Extending RSVP-TE to support Guarantee of Service in MPLS*", Innovative Algorithms and Techniques in Automation, Industrial Electronics and Telecommunications, Springer 9/2007, DOI:10.1007/978-1-4020-6266-7_28 pp 149-154

Finn N., "*Improvement in Ethernet Standards to Further Reduce Latency and Jitter*", Presented at the ODVA Industry Conference and 15th Annual Meeting, Oct. 2012, Stone Mountain, Georgia, USA.

Fischer, A.; Botero, J.; Beck, M.; De Meer, H.; Hesselbach, X., "*Virtual Network Embedding: A Survey*", Communications Surveys & Tutorials, *IEEE* , vol.PP, no.99, pp.1,19, 0 doi: 10.1109/SURV.2013.013013.00155

Prytz, Gunnar. "*A performance analysis of EtherCAT and PROFINET IRT.*", IEEE International Conference on Emerging Technologies and Factory Automation ETFA, 2008.

Laplante, P. *" What Every Engineer Should Know about Software Engineering"*, CRC Press, 2007, ISBN 1420006746.