Libraries and Learning Services

University of Auckland Research Repository, ResearchSpace

Version

This is the Accepted Manuscript version. This version is defined in the NISO recommended practice RP-8-2008 http://www.niso.org/publications/rp/

Suggested Reference

Schlechtendahl, J., Kretschmer, F., Sang, Z., Lechler, A., & Xu, X. (2017). Extended study of network capability for cloud based control systems. *Robotics and Computer Integrated Manufacturing*, *43*, 89-95. doi:10.1016/j.rcim.2015.10.012

Copyright

Items in ResearchSpace are protected by copyright, with all rights reserved, unless otherwise indicated. Previously published items are made available in accordance with the copyright policy of the publisher.

© 2016, Elsevier. Licensed under the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International

For more information, see General copyright, Publisher copyright, Sherpa Romeo.

Extended study of network capability for cloud based control systems

Jan Schlechtendahl[1*], Felix Kretschmer[1], Zhiqian Sang[2], Armin Lechler[1], Xun Xu[2],

[1]Institute for Control Engineering of Machine Tools and Manufacturing Units (ISW), University of Stuttgart 70174 Stuttgart, Germany

[2]Department of Mechanical Engineering, University of Auckland, Auckland 1010, New Zealand

ABSTRACT

Current control systems are limited from a technical viewpoint in areas such as scalability, start-up and reconfiguration time and computational complexity for algorithms. These limitations call for a new concept for control systems to address current and future requirements. It has been suggested that the physical location of the control system be moved from that of the machine to a cloud, i.e. Control System as a service (CSaaS). In this way, the control system becomes scalable and can handle highly complex computational tasks while keeping the process know-hows. Utilizing capabilities of modern Wide Area Network (WAN) and Local Area Network (LAN) the control system can be connected with the rest of the machine, e.g. drives, sensors, devices and HMI. This approach, however, presents new challenges, i.e. the requirement for integration of network, cloud computing and control system expertise. This paper will focus on the requirements of the communication for a cloud based control system.

1. INTRODUCTION

"Intelligence is the ability to adapt to change." - Stephen Hawking

This quote from Stephen Hawking is also valid for production systems today. Only intelligent production systems can meet the requirements of a flexible production of the 21st century with increasing demand for versatility and scalability. Intelligent production systems can be developed only if the control system is intelligent, while current machine controls are not. Being limited in areas like e.g. reconfiguration ability [1], security [2] and computational power [3], the machine control demands for a radically new concept. In recent years, cloud computing have been dramatically changing enterprises and industries in the way of organizing business. Cloud manufacturing adopts the concept of cloud computing, i.e. virtualized manufacturing resources are distributed through Internet as services, and encourages resource sharing and collaboration among medium and small sized enterprises especially [4]. Researcher in cloud manufacturing area mainly focus on the system architecture [5], service management [6], and enabling technologies of cloud manufacturing, such as virtualization of manufacturing resource [7] and service oriented technology [8]. The research of cloud manufacturing is still in its early stage. Only cloud based CAD service [9] is available from the market in the CAD-CAM/CAPP-CNC chain.

The chosen approach of the project of the joined research between the Institute for Control Engineering of Machine Tools (ISW), University of Stuttgart, Germany, and the Department of Mechanical Engineering, University of Auckland, New Zealand, is to provide a control system as a service (CSaaS) from a cloud environment. In this way, the control system becomes scalable and can handle highly complex computational tasks while retaining the process know-hows. Utilizing capabilities of modern Wide Area Network (WAN) and Local Area Network (LAN) the control system can be connected with the rest of the machine, e.g. drives, sensors, devices and HMI. For the owner of the machine, there is no difference to a conventional machine control. This approach, however, presents new challenges, i.e. requirements for integration of network, cloud computing and control system expertise.

Providing control system as a service over networks creates additional challenges resulting from the nature of the IT infrastructure. Communication errors could happen on a communication route and have to be considered by a communication protocol [10]. The errors "repetition", "loss", "insertion", "wrong sequence", "falsification" and "delay" are also relevant for CSaaS and have to be detected and resolved by a communication protocol or even by the

[*] Corresponding author: Tel.: +49-711-685-82464; Fax: +49-711-685-82808; E-mail: jan.schlechtendahl@isw.uni-stuttgart.de

cloud machine control. Although real-time Ethernet gains its popularity in industrial application, it can only be applied in local area networks [11].

As a first test to CSaaS a communication analysis has been done between two servers located in Germany. The results, which support the idea of a CSaaS, were presented on the SPS/IPC/Drives congress in Nuremberg 2013. However, the very short communication path and the limited period (only 1-2 hours) were not able to provide sufficient data to identify the challenges of WAN towards CSaaS.

This paper presents an opportunity analysis for the communication of control data between a machine and cloud-based control. The opportunity analysis is based on two scenarios. For the first scenario, the cloud-based control is located in Stuttgart, Germany whereas the “machine” is located in Auckland, New Zealand. In the second scenario, the cloud-based control is also located in Stuttgart, Germany but the “machine” has been moved to a Google cloud center located in Europe.

In the first section of the paper, requirements of the communication between cloud and machine based on two use cases are defined. Based on the use cases, a network test setup is described in the second section of the paper, which locates the control system’s communication module in Stuttgart and the dummy communication module of a simulated machine in Auckland or at the Google cloud Europe. The test setup is expanded by a monitoring solution to analyze the network behavior. In the third section of this paper, the communication monitoring results will be discussed with respect to parameters, which have a big impact on the production process. As a final section, strategies will be presented for both use cases that could resolve some of the communication challenges of Chapter 3. The paper ends with a conclusion whether a cloud-based machine control for machines could be possible.

2. USE CASES AND COMMUNICATION REQUIREMENTS

As a first step towards a control system as a service, a general understanding of the transferred data between control system and machine has to be developed. It is important to know what type and amount of data is transferred in which cycle time. Further knowledge is needed about the impact of the data on the process results and if the data is part of a control loop. This analysis has been done based on two use cases. One use case is based on a five-axis milling machine located in Stuttgart, the other one is based on a three-axis milling machine in Auckland.

2.1. USE CASE 1: FIVE-AXIS MILLING MACHINE

For the first use case, an Exeron HSC 5-axis milling machine is used. For the Exeron HSC 600 three types of data streams could be identified which have always a two-way communication with the control system:

- Data that is exchanged between spindle/axis drives and the control system. This data includes drive control and status word, setpoint and actual position. The origin and destination of these data is the computerized numerical control (CNC).
- Data that is exchanged between the machine control and I/O terminals. This data is linked to the programmable logic controller (PLC) where e.g., pumps are controlled and information of sensors is evaluated for plausibility checks [12].
- The third data stream is originating and ending in the human machine interface (HMI). Actions taken by the machine user have to be transferred to the machine control (e.g. start NC program) and feedback values (e.g. current line of executed NC program and axis position) have to be transmitted for visualization.

Table 1 shows the data streams corresponding to the amount of data and cycle time. The direction is relative to the machine control. These values are the ones configured by the vendor of the machine.

Table 1: Data streams: amount of data and cycle time of HSC Exeron machine

Type and direction of data stream	Amount of data (Bytes)	Cycle time (ms)
To axis/spindle	46	1
From axis/spindle	96	1
To IO system	50	1
From IO system	52	1

To HMI	Variable	Variable
From HMI	Variable	Variable

A closer look at the transferred data, especially for the I/O terminals, identifies that a cycle time of 1 ms is not necessarily required. The internal bus of the I/O module and the used clamp can expand the reaction time dramatically.

As a second step, control loops, which are depending on status values resulting from the machine in the machine control, have to be identified. For the Exeron HSC 600 machine the most critical control loops are resulting from the axis and the CNC system. The CNC is performing the following:

- Check if the actual position is within a monitoring window to the commanded position. If this is not the case, an error is set. This behavior results in the requirement to either deactivate or increase the monitoring window or do the comparison within a short cycle time, for example by shifting this checking procedure closer to the machine.
- Check if changes to the drive control are executed according to the expectations of the CNC and correctly reported by the drive over the drive status. A possible solution would either be a deactivation of the checking procedure or always reporting a good drive status to the CNC.

For the PLC control loops are normally not critical since most components controlled by the PLC are activated and then the actual status is checked before further steps are initialized (e.g. activate pump over output clamp and check over input clamp if pump started working). Adding extra delays in these loops will decrease the machine performance but will have no influence on the quality of the work piece, for example.

Since HMIs are mostly windows operating system-based, no real-time requirements arise. A reaction on the user input is expected within 200 milliseconds. If the reaction time is longer, the user satisfaction is decreased, but without influence on the work piece quality.

2.2. Use Case 2: Three-axis Milling Machine

As the second use case, a Sherline 2000 series 3-axis milling machine is evaluated. A real-time Ethernet fieldbus, EtherMAC, is used to control the motors and I/O points. As the control of the machine tool is moved to the cloud, however, a local control of the machine tool is still needed, especially for setting up coordinate and inspection systems. A local control system is proposed as shown in Figure 1.

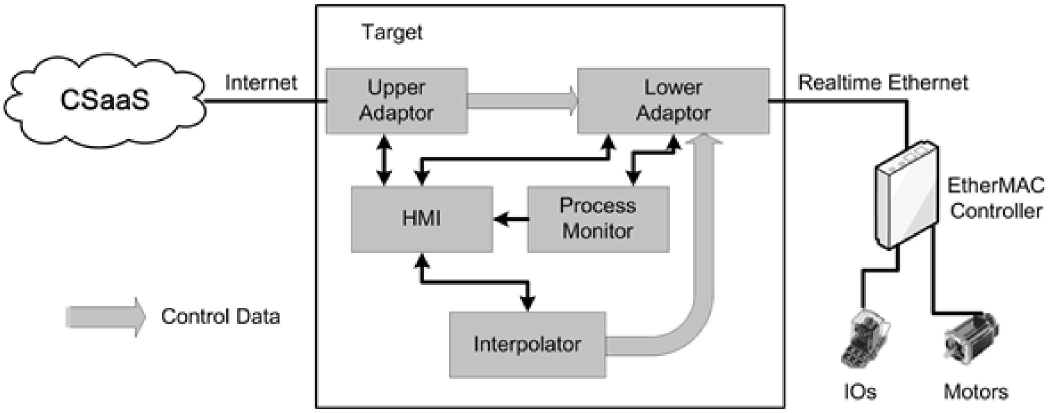

Figure 1: System structure of the local control

The upper adaptor receives the network packages from the cloud, which contains the interpolated data, and extracts the motor control command for every cycle, which will then be “translated” by the lower adaptor following EtherMAC protocol. The lower adaptor sends and receives packages to and from the EtherMAC controller to control the motors, set I/O outputs and reads in axis feedback and I/O inputs. The local interpolator takes charge of some motion control functions, for instance, manual operation and setting coordinate systems. The local HMI receives instructions from the local operator and notifies the lower adaptor to execute control commands from either the upper adaptor or the local interpolator. It also displays information useful for local operators including current coordinate information, feed rate, spindle speed, tool information, each I/O status and maintenance information. The process monitor keeps an eye on the communication and machining process. A serial number is attached to every package transmitted from the cloud to the local control. When an error occurs, the process monitor will stop the machining and send the serial number back to the cloud, based on which the error can be located in the part program. The data package will be discarded thereafter.

Data stream requirements are shown in Table 2. The amount of data comes from EtherMAC protocol and only includes the essential data for controlling the machine tool. The data for controlling the axes, the spindle and I/O terminals and the feedback information from the machine are transmitted every four millisecond. HMI data has no real-time requirement. Some information will be transmitted only when necessary. The data transmitted back to the cloud will have no effect on the control. However, as the CNC in the cloud has all the feedback information from the machine, if the difference between command data and feedback data is beyond a certain threshold, an error will be triggered.

Table 2: Data streams: amount of data and cycle time of Sherline machine

Type and direction of data stream	**Amount of data (Bytes)**	**Cycle time (ms)**
To machine:		
-Axis/Spindle/IO/Serial Number	53	4
-HMI	Variable	Variable
To cloud:		
-Axis/IO	18	4
-HMI	Variable	Variable

A conclusion from this chapter is that every machine whose control should be provided as a service has to be analysed in regard to the transferred data and control loops that exist between machine control and machine tool.

3. TEST SETUP AND MONITORING SOLUTION

As a next step on the path to a cloud based control system, the influence of the Wide Area Network (WAN) and the Local Area Network (LAN) has to be evaluated. For this purpose, a communication test setup has been developed as shown in Figure 2. One part of the test setup – the “cloud communication module” – located in Stuttgart, Germany, is creating the data that is transferred from the cloud to the machine and transmits it over WAN and LAN. The machine communication module, which receives the data from Stuttgart, is located either in Auckland, New Zealand or in Google cloud center located in Europe. In Auckland and at the Google cloud center the data is received, logged, and as a second step, the “machine communication module” creates and transmits the data from the machine to the cloud system. Both communication channels – from Stuttgart and to Stuttgart – could be configured to use different connections as User Data Protocol (UDP), Transmission Control Protocol (TCP) and Websocket Protocol [13].

Both communication modules run on non-real-time operating systems. This does not necessarily guarantee the deterministic transmission and reception of telegrams in the cycle time of the use cases. To achieve a similar behavior, the communication modules have an internal timer, which allows them to recognize the elapsed time since the last execution. If cycles are missed, the communication modules regain the lost cycles by multiple execution of the send and receive procedure.

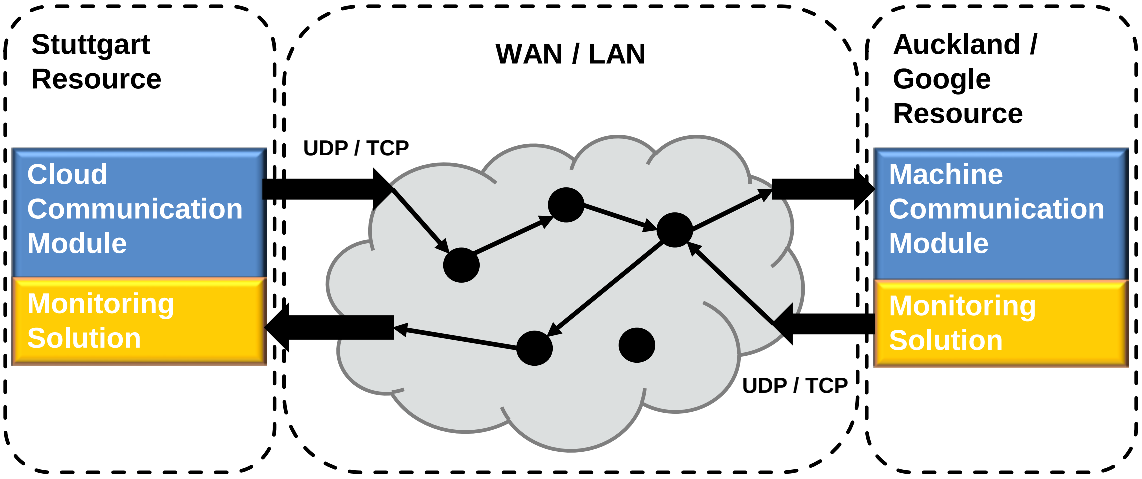

Figure 2: Test setup communication

For monitoring the network behavior, a monitoring solution has been integrated in the communication modules. The monitoring solution includes the following information in the transmitted data:

- Cloud Communication Module: Consecutive counter which increases every execution cycle
- Cloud Communication Module: Time which is currently valid in the operating system
- Machine Communication Module: Consecutive counter of received telegrams
- Machine Communication Module: Time which is currently valid in the operating system
- Machine Communication Module: Consecutive counter of sent telegrams

After receiving a telegram, all information is stored in log files. Further, the monitoring solution is evaluating the following on-line for an easier analyzes:

- Maximal latency in a defined time interval
- Minimal latency in a defined time interval
- Average latency in a defined time interval
- Number of maximal consecutive telegram losses in a defined time interval (UDP only)
- Sum of telegrams lost in a defined time interval
- Length of queue (TCP only)

Based on the information it is possible to evaluate which challenges have to be expected for a cloud-based control located in either Germany and a machine based on the counterpart.

4. Monitoring results of use cases

For the two use cases the following measurements have been created by the test setup. Both measurements have used the amount of data and the cycle time of the use cases described above.

4.1. Scenario 1: Communication Between Stuttgart And Auckland

In the first scenario it has been discovered, that the capability to send data is limited in Auckland and is not able to transmit 100 Bytes of data in millisecond intervals without heavy telegram losses. To reduce the impact of a limited transmission capability the bigger amount of data is always send from Stuttgart to Auckland and the smaller amount of data is always received.

4.1.1 Use Case 1: Five Axis Milling Machine

For the first measurement, 85752629 UDP packages have been transmitted with a payload of 100 Byte to Auckland and 50 Byte to Stuttgart. On the path to Auckland 1% (863924 telegrams) of the data was lost. On the way back to

Stuttgart 2.9% (2488668 telegrams) of the data was lost. In average, the round trip time was 0.3174 seconds, but with big peaks as shown in Figure 3. Quite often, the average round trip time (RTT) scaled on 10s time intervals increased up to 2 seconds with the overall peak of 17 seconds.

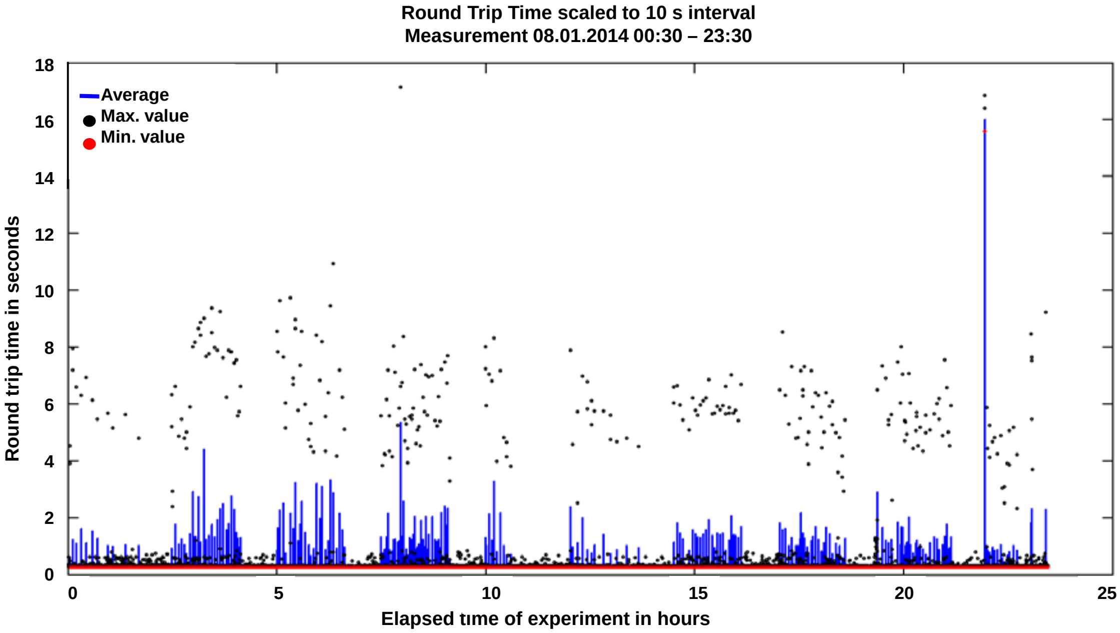

Figure 3: Scenario 1 - Use Case 1 - UDP - Round Trip Time

Looking at the consecutive telegram losses in Figure 4 it can be seen that the number of losses in certain 10-second intervals increased almost up to 10000 telegrams.

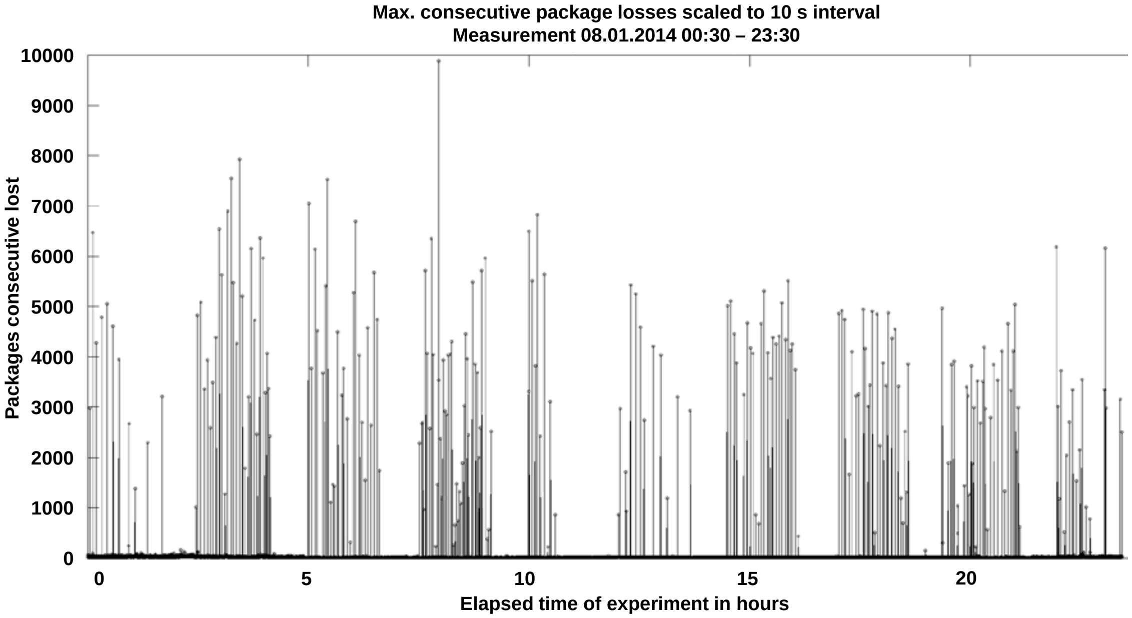

Figure 4: Scenario 1 - Use Case 1 - UDP - Consecutive telegram losses

Further – looking at the TCP communication – it can be stated that the TCP communication is not usable for a cloud to machine communication in this use case between New Zealand and Germany. Even though TCP has quality of service mechanisms integrated, which result in no telegram losses, the transmision of telegrams is slower as with UDP. In the measurement it was not possible to transmit more than 305 telegrams per second. In certain time intervals, where not enough bandwidth is available, the sent queue in Auckland is increasing which also influenced the RTT. As shown in Figure 5 the RTT at the end of the measurement increased up to 3500 seconds.

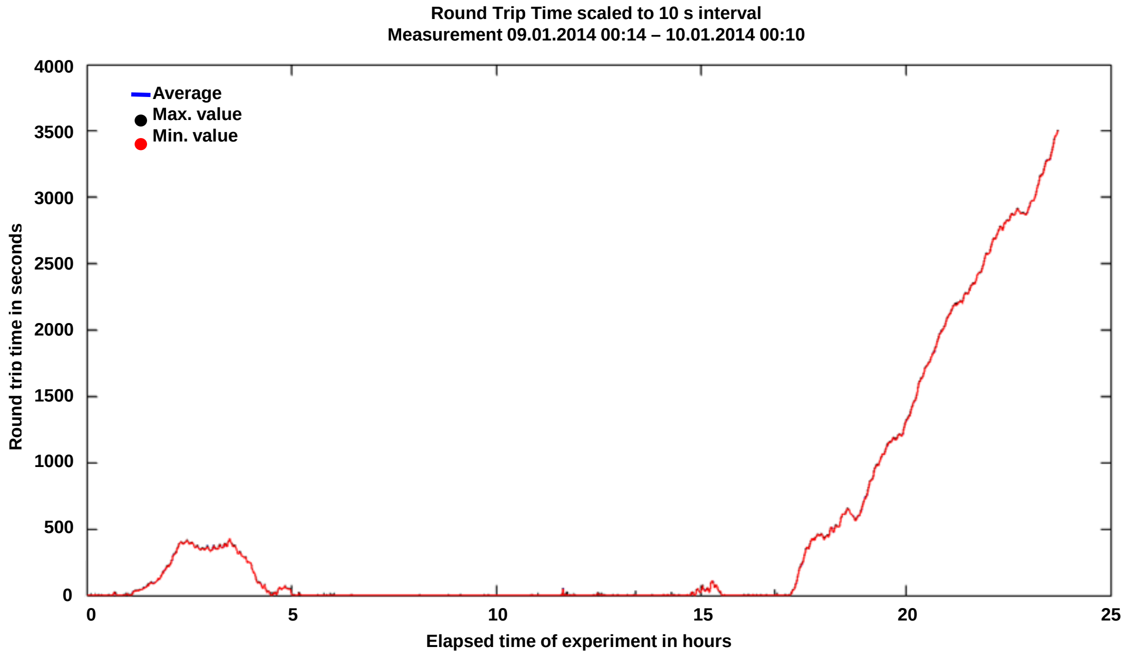

Figure 5: Scenario 1 - Use Case 1 - TCP - Round Trip Time

4.1.2. *Use Case 2: Three-axis Milling Machine*

During a consecutive test of about 24 hours, 27015516 UDP packages were sent, among which 2.64% (715237) were lost. Similar to the first use case, most package loss happened during transmission from Auckland to Stuttgart, which is 2.31% compared to 0.33% from Stuttgart to Auckland. The peak number of consecutive package losses in 10s intervals occurred 21 hours after the start of the test, which is more than 450 packages. Nearly at the same time, the peak package loss from Auckland to Stuttgart was witnessed, which is nearly 2500 packages. The overall average RTT was 0.3466s, and the maximum RTT was about 20s. However, most of the maximum RTT is less than 11s. It also seems that there was a period of loss between better performance and worse performance. In the transmission point of view, UDP connection can transmit 313 packages per second, which fulfills the second use case. However, the package loss is an essential issue. Missing packages will fail the machining and even cause damage to the machine tool.

In another consecutive 24-hour test, 27000269 TCP packages were transmitted. The maximum RTT was about 85s, and a great number was above 10s. Due to the capacity of TCP communication, there were long queues of packages waiting to be transmitted from time to time, even more than the packages needed for machining some small parts, which causes the whole system's deficiency.

4.2. *Scenario 2: Communication Between Stuttgart And Google Cloud Center Europe*

4.2.1 *Use Case 1: Five Axis Milling Machine*

Figure 6 shows the measurement of 100byte of data via a websocket connection between a local client in Stuttgart and a remote server at the Google Cloud Center Europe. The cycle time of each generated package with a random payload was 1ms. The 59795451 transferred packages had an average RTT of 42.91ms with maxima peaks at 150ms. The measurement resulted in the collection of packages from the Cloud Center to Stuttgart (the sender received several packages at the same time) and therefore increased the RTT for the each telegram within a collection from the minima at 25ms to 150ms at the maxima RTT.

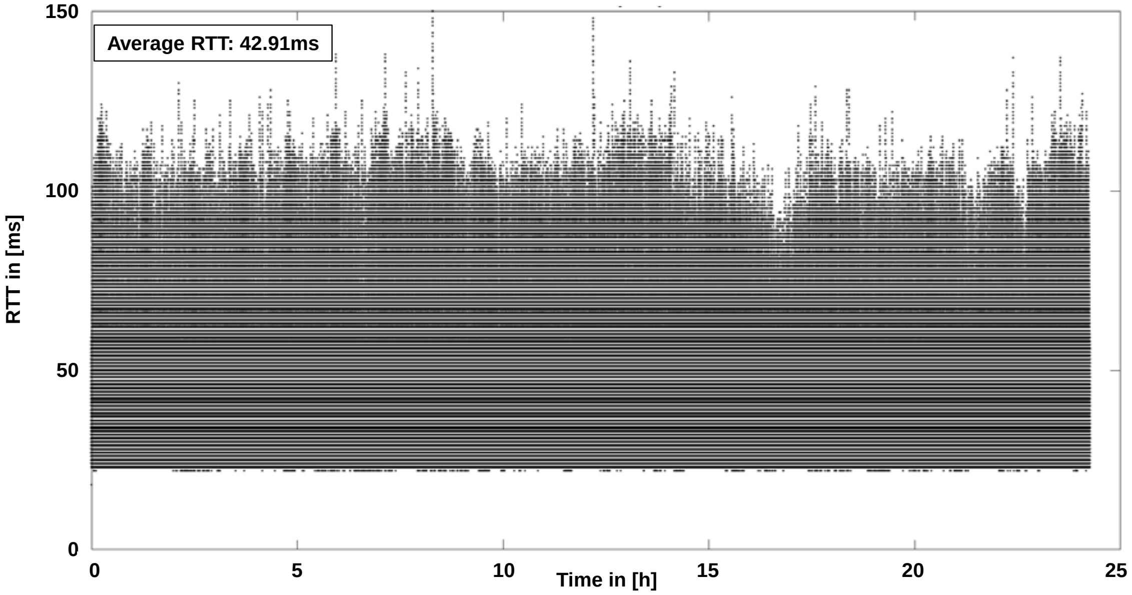

Figure 6: Scenario 2 - Use Case 1 – Websocket - Round Trip Time

4.2.3. Use Case 2: Three-axis Milling Machine

The measurement of the second use case sent 19290892 number of packages with an average RTT of 47.16ms as seen in Figure 7. Depending on the use of parts of the connection by other clients the measurement shows maxima peaks up to over 3500ms. According to the first use case packages were also received in collections and therefore increased the RTT for each package within one collection.

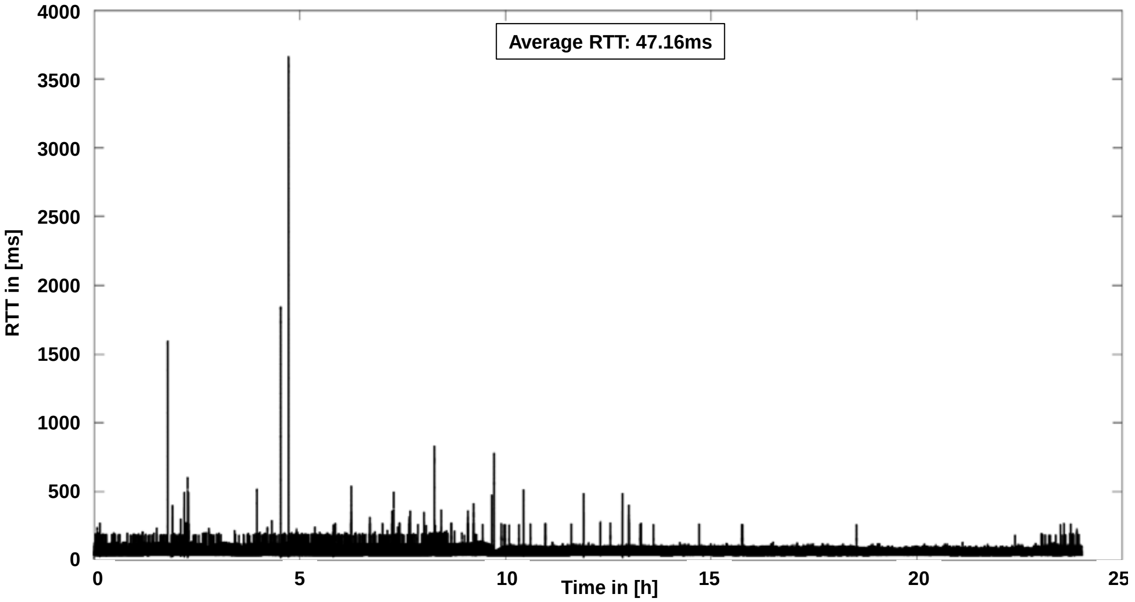

Figure 7: Scenario 2 - Use Case 2 – Websocket - Round Trip Time

5. Strategies for Communication Challenges

It has been shown in the previous sections that the TCP connection is not applicable for a control system as a service (CSaaS) between Auckland, NZ and Stuttgart, Germany. Missing bandwidth capability results in queues and increased RTT, which are not acceptable for both use cases.

UDP connections have an average telegram loss of 2.64% - 3.9% but with peaks up to 10000 consecutive telegram losses. This means that the machine is not receiving telegrams for almost 10 seconds. This might be solved through buffers located in the communication modules. To refill the buffers, mechanisms need to be implemented that stop the machine and allow the buffers to be refilled. Further, the machine communication module - depending on the type of data transferred - might interpolate single missing telegrams.

Another possible way to deal with package loss is adding serial numbers to every package. If receiving a non-consecutive data package, the receiver can request packages with certain serial numbers. However, if the network is encountering performance issues, the successive requesting of missing packages may increase the burden on the network and make the performance even worse.

Websocket connections between Stuttgart and a Google Cloud Center in Europe show better results: The average RTT is usable for CSaaS. Problems with missing telegrams did not occur through the design of the websocket protocol. Only the maximum peaks of the RTT of 3500ms during use case two is not acceptable.

Network performance monitoring mechanism and strategies for dealing with the performance issue may be needed by both CSaaS and the machine tool. When performance deterioration is detected or predicted, the machine tool can make decisions based on strategies including adaptively pausing the machining in the middle of executing a toolpath or at the connection point of adjacent toolpaths, which also requires additional information. When the network performance restores and enough packages are received and stored in local cache, the machining could be resumed.

6. Summary

In this paper two use cases based on two milling machines have been described. The data transferred between control system and machine in these use cases has been analyzed. Then the data was used to analyze if a control system as a service (CSaaS) is possible. For doing this, a test environment was set up where the data according to the use cases could be transferred between Auckland, New Zealand and the Google Cloud Center in Europe and Stuttgart, Germany. The data transfer has been monitored and analyzed in the last part of the paper. A conclusion can be stated that CSaaS between New Zealand and Germany is not possible, since the network challenges are too big. The control system should be located closer to the machine. CSaaS between the Google Cloud Center in Europe and Germany is possible for slow cycle times and depends on the usage of the connections by other clients. The Google Cloud platform is not suitable for control systems that require high performance connections. Due to technical conditions (e.g. load balancer, routing mechanisms, etc.) the transferred packages are received in collections by the local clients. Slower cycle times will probably reduce the number of collections but result in less performance for the control system.

The existing general TCP/UDP/Websocket protocol is not suitable for interpolated data to be transmitted through WAN connection. The unpredictability of the WAN connection requires additional mechanisms and strategies to be developed for both CSaaS and machine tool.

Acknowledgements

The research presented in this paper is resulting from the project “pICASSO” which is funded by the German Federal Ministry of Education and Research. The authors are grateful for the financial support of China Scholarship Council - University of Auckland joint scholarship.

References

[1] Lorenzer T, Weikert S, Wegener K. Mit Rekonfigurierbarkeit gewinnt der Anwender (engl.: gain users with reconfiguration ability). Eidgenössische Technische Hochschule Zürich, Institut für Werkzeugmaschinen und Fertigung, Zürich, 2010.

[2] Birkhold M, Verl A. Post-Stuxnet: Sicherheitslücken bedrohen weiterhin Produktionsanlagen (engl.: post-Stuxnet: security bugs putting production sites on risk), ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb 04/2011, p. 237-240

[3] Keinert M, Verl A. System Platform Requirements for High-Performance CNCs, Proceedings of FAIM 2012 22nd International Conference on Flexible Automation and Intelligent Manufacturing, Helsinki, Finland, 06/ 2012.

[4] X. Xu: "From cloud computing to cloud manufacturing", Robotics and Computer-Integrated Manufacturing, Vol.28, No.1, pp.75-86, 2012.

[5] X. Vincent Wang and X. W. Xu: "An interoperable solution for Cloud manufacturing", Robotics and Computer-Integrated

Manufacturing, Vol.29, No.4, pp.232-247, 2013.

[6] *W. He and L. Xu: "A state-of-the-art survey of cloud manufacturing", International Journal of Computer Integrated Manufacturing, pp.1-12, 2014.*

[7] *N. Liu and X. Li, "A resource virtualization mechanism for cloud manufacturing systems", Enterprise Interoperability, Springer, 2012.*

[8] *O. F. Valilai and M. Houshmand: "A collaborative and integrated platform to support distributed manufacturing system using a service-oriented approach based on cloud computing paradigm", Robotics and Computer-Integrated Manufacturing, Vol.29, No.1, pp.110-127, 2013.*

[9] *http://www.autodesk.com/360-cloud/ (accessed February 27, 2014)*

[10] *M. Felser: "Real-Time Ethernet - Industry Prospective", Proceedings of the IEEE, Vol.93, No.6, pp.1118-1129, 2005.*

[11] *N.N. Sicherheitsrelevantes Konstruieren von Druck und Papierbearbeitungsmaschinen – Elekktrische Ausrüstung und Steuerungen (engl. Safety relevant construction of printing and papercutting machines – eletrical equipment and control. BG Druck und Papierverarbeitung. ed. 06/2004. p. 79.*

[12] *M. Weck C. Brecher: Werkzeugmaschinen 4 – Automatisierung von Maschinen und Anlagen (engl.: machine tools 4 – automation of machines and sites). 6th ed. Berlin/Heidelberg: Springer; 2006.*

[13] *Internet Engineering Task Force (IETF): The WebSocket Protocol RFC 6455; http://tools.ietf.org/html/rfc6455 (accessed March 27 2015)*