Towards the Future Internet – A Survey of Challenges and Solutions in Research and Standardization*

Thomas Dreibholz, Erwin P. Rathgeb
University of Duisburg-Essen, Institute for Experimental Mathematics
Ellernstrasse 29, 45326 Essen, Germany
{thomas.dreibholz,erwin.rathgeb}@uni-due.de

1. Introduction

The basic intention for the development of the Internet has been the simple and inexpensive interconnection of nodes to provide services like file download or e-mail. However, due to its huge growth and popularity, the classical protocols – which still constitute the basis of today's Internet – have reached their limits of scalability and functionality. Lots of research has been performed in order to overcome these restrictions. On our poster, we would like to give a survey of the challenges on Network, Transport, Session and Application Layer, as well as an outline of solutions which will constitute – from the current perspective in 2007 and with regard to the standardization progress of the IETF – the basis of the Internet in 2022.

2. Network Layer Challenges

On the Network Layer, a too small IPv4 address space as well as its inefficient partitioning lead to overly long routing tables and expensive routing. IPv6 [2] has been designed to overcome these limitations. A particular feature of IPv6 is that each endpoint possesses multiple addresses (possibly with different scopes). Furthermore, the existing IPv4 infrastructure will still have a long lifetime – and each endpoint will also have an IPv4 address unless (almost) every host has been made IPv6-capable. That is, future hosts will be *multi-homed*, i.e. reachable under different addresses and even Network Layer protocols. This property – and its utilization for a better service – becomes a challenge for the Transport Layer (see section 3).

A particularly important IPv6 feature is to easily allow for site-renumbering by its auto-configuration. A site renumbering means to change the prefix (i.e. the network address). Unlike IPv4 – where assigned network addresses are transferred on provider changes, leading to large backbone routing tables – this also puts challenges on the Transport Layer.

Recent advances on the Network Layer are not restricted to the routed protocols, also the routing procedure itself is improved. While the foundation of the Internet has been stateless packet forwarding due to scarce resources, future generations of routers will be powerful enough to perform *flow routing* [1]. Having a state for each routed flow, applying Quality of Service (QoS) [3] mechanisms gets easy. A novel approach for a simple QoS mechanism has been proposed by us in [7, 11]: instead of setting up reservations by complex QoS signalling procedures, we simply remember a small set of flow identities. These remembered flows will be in the focus of packet discard on network overload. All other flows should not suffer from packet loss.

* Parts of this work have been funded by the German Research Foundation (Deutsche Forschungsgemeinschaft).

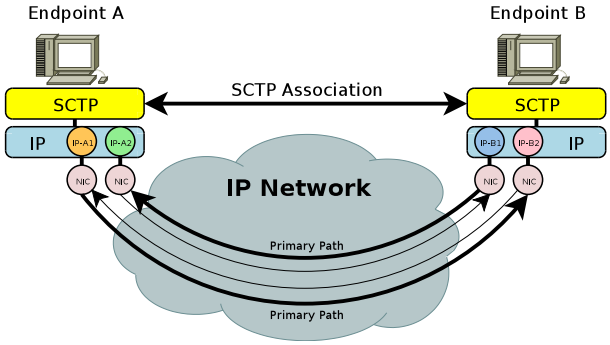

Figure 1. Multi-Homing with SCTP

3. Transport Layer Challenges

Using TCP for the transport of upper-layer data, a connection between two endpoints – given by a selected pair of Network Layer addresses – is established. Upon change of one of the addresses (e.g. due to a site renumbering), the connection is broken. Furthermore, if the endpoints are interconnected by two distinct networks, the connection only uses a single network for the transport (given by the address pair). If this network fails, the connection is also broken.

In order to overcome this challenge, the SCTP (Stream Control Transmission Protocol) [15,16] – a connection-oriented and reliable Transport Layer protocol – has been designed and is now supported by all major operating systems. Its most important feature is multi-homing (as illustrated in figure 1): as long as there is at least one possible path between two endpoints, a connection stays usable. Furthermore, SCTP allows for address changes (Add-IP) [17]. In particular, applying Add-IP on a connection even allows for its establishment in an IPv4-only network, later adding IPv6 addresses during transition to IPv6 and finally removing the obsolete IPv4 addresses – without breaking the association or even bothering the upper layers! The main application of Add-IP is to add or remove additional links for redundancy reasons and to support endpoint mobility for long-lasting transport connections.

Furthermore, SCTP provides the preservation of message frames and multi-streaming. The multi-streaming feature allows for multiplexing different data flows over a single transport association, which is in particular useful for the transport of VoIP/multimedia trunk data. Furthermore, it is possible to turn off packet retransmission on a per-message basis (PR-SCTP option). However, unlike UDP, there is always congestion control for the transport association. Secure-SCTP (S-SCTP) [14,18], a further optional extension which has been developed by us, provides per-message encryption and authentication.

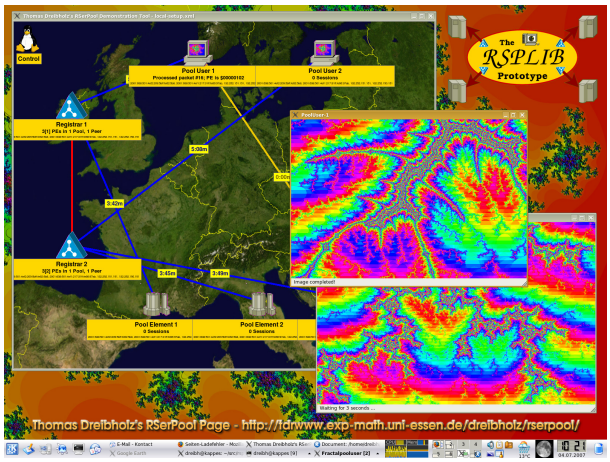

Figure 2. Our RSerPool Prototype Demo

4. Session Layer Challenges

While SCTP already addresses network path redundancy, a service is also broken if the server itself fails. Due to an increasing amount of critical services, it has been necessary to design an unified and application-independent approach to provide server pool management, session handling and the support of a session failover between servers. This approach by the IETF – Reliable Server Pooling (RSerPool) – in particular defines the IETF's first Session Layer protocol [6].

Server redundancy leads to load distribution and load balancing, which are also covered by RSerPool [10, 13]. But in strong contrast to already available solutions in the area of GRID and high-performance computing, the fundamental property of RSerPool is to be "lightweight", i.e. it must be usable on devices providing only scarce memory and CPU resources (e.g. embedded systems like telecommunications equipment). This property restricts the scope of RSerPool to the management of pools and sessions only, but on the other hand supports a very efficient realization [9]. Any further functionalities are considered to be application-specific and may be provided by upper layers. In particular, the application may use arbitrary failover procedures. However, due to its simplicity, RSerPool provides client-based state sharing [4] for failover handling: a PE may send its current session state to the PU in form of a state cookie. The PU stores the latest state cookie and provides it to a new PE upon failover. Then, the new PE simply restores the state described by the cookie. Cryptographic methods can ensure the integrity, authenticity and confidentiality of the state information.

5. Application Layer Challenges

Based on RSerPool, it is possible to provide a critical service by using unreliable components. This means that a server pool can consist of inexpensive, off-the-shelf PC components. Designing applications directly with server redundancy by RSerPool in mind, this may lead to a significantly improved cost-benefit ratio of services. In the extreme case, a service could (mainly) be provided by currently idle PCs – which have to leave the pool immediately if required otherwise. During our poster presentation, we will illustratively demonstrate this concept – as well as the multi-homing features of SCTP – by a demo presentation of our Open Source RSerPool reference implementation RSPLIB [5, 12]! A screenshot of our demo system [8] is depicted in figure 2.

References

[1] J. L. Adams, A. IJsselmuiden, and L. Roberts. An advanced QoS protocol for real-time content over the internet. In *Proceedings of the 13th International Workshop on Quality of Service (IWQoS)*, pages 164–177, Passau/Germany, June 2005.

[2] S. Deering and R. Hinden. Internet Protocol, Version 6 (IPv6). Standards Track RFC 2460, IETF, Dec. 1998.

[3] T. Dreibholz. Management of Layered Variable Bitrate Multimedia Streams over DiffServ with Apriori Knowledge. Masters Thesis, University of Bonn, Institute for Computer Science, Feb. 2001.

[4] T. Dreibholz. An Efficient Approach for State Sharing in Server Pools. In *Proceedings of the 27th IEEE Local Computer Networks Conference*, pages 348–352, Tampa, Florida/U.S.A., Oct. 2002. ISBN 0-7695-1591-6.

[5] T. Dreibholz. Thomas Dreibholz's RSerPool Page, 2006.

[6] T. Dreibholz. *Reliable Server Pooling – Evaluation, Optimization and Extension of a Novel IETF Architecture*. PhD thesis, University of Duisburg-Essen, Faculty of Economics, Institute for Computer Science and Business Information Systems, Mar. 2007.

[7] T. Dreibholz, A. IJsselmuiden, and J. L. Adams. An Advanced QoS Protocol for Mass Content. In *Proceedings of the IEEE Conference on Local Computer Networks 30th Anniversary*, pages 517–518, Sydney/Australia, Nov. 2005. ISBN 0-7695-2421-4.

[8] T. Dreibholz and E. P. Rathgeb. An Application Demonstration of the Reliable Server Pooling Framework. In *Proceedings of the 24th IEEE INFOCOM*, Miami, Florida/U.S.A., Mar. 2005. Demonstration and poster presentation.

[9] T. Dreibholz and E. P. Rathgeb. Implementing the Reliable Server Pooling Framework. In *Proceedings of the 8th IEEE International Conference on Telecommunications*, volume 1, pages 21–28, Zagreb/Croatia, June 2005. ISBN 953-184-081-4.

[10] T. Dreibholz and E. P. Rathgeb. On the Performance of Reliable Server Pooling Systems. In *Proceedings of the IEEE Conference on Local Computer Networks 30th Anniversary*, pages 200–208, Sydney/Australia, Nov. 2005. ISBN 0-7695-2421-4.

[11] T. Dreibholz, A. Smith, and J. L. Adams. Realizing a scalable edge device to meet QoS requirements for real-time content delivered to IP broadband customers. In *Proceedings of the 10th IEEE International Conference on Telecommunications*, volume 2, pages 1133–1139, Papeete/French Polynesia, Feb. 2003. ISBN 0-7803-7661-7.

[12] T. Dreibholz and M. Tüxen. High Availability using Reliable Server Pooling. In *Proceedings of the Linux Conference Australia*, Perth/Australia, Jan. 2003.

[13] T. Dreibholz, X. Zhou, and E. P. Rathgeb. A Performance Evaluation of RSerPool Server Selection Policies in Varying Heterogeneous Capacity Scenarios. In *Proceedings of the 33rd IEEE EuroMirco Conference on Software Engineering and Advanced Applications*, Lübeck/Germany, Aug. 2007.

[14] C. Hohendorf, T. Dreibholz, and E. Unurkhaan. Secure SCTP. Technical Report Version 02, IETF, Individual Submission, June 2007. draft-hohendorf-secure-sctp-02.txt, work in progress.

[15] A. Jungmaier. *Das Transportprotokoll SCTP*. PhD thesis, Universität Duisburg-Essen, Institut für Experimentelle Mathematik, Aug. 2005.

[16] A. Jungmaier, E. P. Rathgeb, and M. Tüxen. On the Use of SCTP in Failover-Scenarios. In *Proceedings of the State Coverage Initiatives 2002, Volume X, Mobile/Wireless Computing and Communication Systems II*, volume X, Orlando, Florida/U.S.A., July 2002. ISBN 980-07-8150-1.

[17] M. Ramalho, Q. Xie, M. Tüxen, and P. Conrad. Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration. Technical Report Version 15, IETF, Transport Area Working Group, Mar. 2006. draft-ietf-tsvwg-addip-sctp-15.txt, work in progress.

[18] E. Unurkhaan. *Secure End-to-End Transport - A new security extension for SCTP*. PhD thesis, University of Duisburg-Essen, Institute for Experimental Mathematics, July 2005.