# Content Delivery Network Interconnect

Andrej Binder*
Institute of Computer Engineering and Applied Informatics
Faculty of Informatics and Information Technologies
Slovak University of Technology in Bratislava
Ilkovičova 2, 842 16 Bratislava, Slovakia
andrej.binder@stuba.sk

## Abstract
Content Delivery Networks have been the driving force of globally scalable services on the Internet for almost two decades. They have always been developed and implemented as overlay networks run by companies independent from the network providers. In the recent years there has been a significant pressure to make it possible for content providers and telecom operators to be able to enter the content delivery market. This move is a sensible one because they have access to information about their network topologies that puts them into an unique position to distribute and deliver content more efficiently than the over the top players.

The only problem is that these Telco companies usually only have a local coverage. This means that a protocol has to be developed to make it possible for multiple local CDN's to interconnect into a single CDN Federation witch much broader coverage. The design of such a protocol and related research are the goals of this thesis.

## Categories and Subject Descriptors
C.2.1 [**Computer communication networks**]: Network Architecture and Design—*Distributed Networks*; C.2.2 [**Computer communication networks**]: Network Protocols

## Keywords
Telecommunications, Internet, Content Delivery Networks, Content Delivery Network Interconnection, Network Protocol, Network Architecture

*Recommended by thesis supervisor: Assoc. Prof. Ivan Kotuliak
Defended at the Faculty of Informatics and Information Technologies, Slovak University of Technology in Bratislava on July 12, 2017.



Binder, A. Content Delivery Network Interconnect. Information Sciences and Technologies Bulletin of the ACM Slovakia, Vol. 9, No. 2 (2017) 38-44

## 1. Introduction
In today's Internet majority of traffic is transferred via unicast[1]. Unicast means that traffic is transferred directly from a source host to a destination host without any multiplexing on the way[7] [8]. The reason for the popularity of unicast communication is rooted in the fact that it is the most versatile and easy to implement method available today. Its problem is that it is inefficient. This inefficiency is especially significant in cases of popular content that has a single source and a large group of destinations that it needs to be delivered to.

The obvious solution would be to use a delivery method other than unicast for this type of content. One example of such use is multicast[20]. It is a method that makes it possible to send a single packet of data in such a way that allows the network infrastructure to multiplex it and deliver it to large group of destinations. The mayor problem with this approach is that it has never managed to gain significant popularity in the scope of public Internet. This means that it can be used for solutions that are locally scoped, but cannot be practically used for applications that require global coverage.

In the recent years the most popular way of overcoming the issue of unicasting popular content has been the use of Content Delivery Networks (CDN's)[2][3]. These are application layer networks that are specifically designed to intelligently cache and multiplex content[4][5]. The basic principle of their operation is based on adding multiple intermediary nodes between the source of the content and the group of destinations that are interested in receiving it[6]. This enables multiplexing without requiring any changes to network infrastructure. It also makes it possible to cache content on the nodes because they are not limited in storage as much as common network devices are. Content Delivery Network nodes are based on full-fledged server hardware as opposed to classical network devices that are build on hardware specialized for routing and switching.

The problem with Content Delivery Networks is that they are based on proprietary protocols and centrally managed[15]. This means that every Content Delivery Network is completely autonomous with no way of exchanging resources with others[14]. The most reasonable way of overcoming this limitation would be to design a standardized interconnection protocol that would clearly define the language in which CDN's could communicate between each other [9]. The creation of such a protocol is the main goal of this dissertation.

This document is split into six chapters, each of them documenting a separate stage of work done on the thesis. You are now reading the first chapter of the thesis which includes the introduction to the thesis and the overview of the structure of the document. The second chapter includes the analysis of the state of the art in the problem area. The third chapter clearly specifies the goals of the thesis. The fourth chapter describes the design of the Content Delivery Network Interconnection protocol. The fifth chapter documents the verification process performed in relation to the thesis. The sixth chapter includes the conclusion of the work, highlighting the original contributions of the author of the thesis.

## 2. State of The Art

### 2.1 Problem Impact Analysis

New standards and technologies are always developed ahead of the time when they are to be used. This means that it is important to look at various predictions and forecasts when analyzing their possible use cases and design requirements.

The Cisco Visual Networking Index (VNI) Forecast[1] clearly indicates that the Video traffic type will very quickly become the most significant portion of global Internet traffic. This is the one of the main reasons why CDN‘s (as the most efficient methods of optimizing video traffic) are more relevant now than ever before and will only keep on being even more relevant in the future.

### 2.2 CDN Technology

Content Delivery Networks (CDN) are systems for the efficient delivery of digital objects (e.g. files with multimedia content as video on demand or other file types) and multimedia streams (e.g. live television streams) over IP networks to many end points and viewers[11]. Typically, a CDN consists of one or more servers that deliver the digital objects and/or streams, and a management/control system. The management/control system takes care of content distribution, request routing, reporting, metadata and other aspects that make the system work.[5] [19]

Currently implemented CDNs are based on a widespread network of distribution nodes controlled by a functional entity taking care of content distribution decisions[16] [18]. The control functional entity keeps track of all content locations, manages distribution amongst the distribution nodes and/or clusters and also decides which distribution node should serve a client request.

The distribution tree representing the flow of content data blocks from its source to its destination can be visualized as a tree graph rooted in its source. The leaves of such a tree are the destinations. The traffic flow of the same traffic would take the same path if CDN was not used but the difference is that the data does not need to be sent over the higher-level links (the ones closer to the root) if CDN is used. This means that in if there were a CDN with one source, three nodes and three hundred destinations evenly distributed amongst the three nodes then the data would only have to travel once between the source and the first node if CDN is used. It would have to travel three hundred times over that link if CDN was not used[10].

### 2.3 CDNI Technology

Every CDN may have a specific network footprint in which it can deliver content effectively. This footprint depends on the locations of distribution nodes that the CDN controls. It is often inefficient to build new distribution nodes that are not in the region in which the company running the CDN is operating. A solution is to interconnect two CDN‘s and share existing infrastructure for content delivery. Interconnection of CDNs is achieved by interconnecting the centralized functional entities that represent the logic behind the decisions in each CDN. This is called CDNI or Content Delivery Network Interconnection.

CDNs are in general autonomous networks offering different services to their users. A CDN’s primary function is to optimize content distribution and delivery. In addition to this primary function many CDNs chose to implement various other capabilities like content manipulation, digital rights management (DRM), intelligent handling of segmented content and others.

Because of this fact, different feature sets need to be defined. A basic feature set is a set of features that are mandatory for every CDN that wants to take part in a CDN federation (a group of interconnected CDN‘s). This set of features is described in the next section. The following section describes a set of features that a CDN may optionally implement.

To simplify the design and description of this relationship a simple semantic model was designed. This model defines two roles in which each CDN can operate. The first mode is one in which a CDN is redirecting requests of its clients to another CDN‘s nodes for delivery to those clients. A CDN operating in this mode is called upstream CDN (or uCDN in short). The CDN that is serving content for the clients of another CDN is called the downstream CDN (or dCDN in short).

The basic features that every CDN interconnection requires in order to work are these:

- Interconnection control
- Request Routing
- Content Distribution
- Footprint Exchange
- Metadata Exchange
- Content Status exchange
- Report Exchange
- Capability Exchange

Capability exchange is a functionality allowing a dCDN to inform an uCDN about the features that it is capable to offer to the uCDN in addition to basic content distribution and delivery. Some of these features are metadata defined reporting, content integrity control, content adaptation, multi-segment content support, content access control, content security, content DRM and others.

## 3. Problem Area Definition and Thesis Goals

The main goal of this thesis is to create a protocol for interconnection of Content Delivery Networks and position it in such a way that the inventions created within will have the best possible chance of being implemented in production environments all over the world.

The main goals of the thesis are as follows:

- Design a CDN Interconnection architecture and disseminate it via contributions to the ETSI CDNI standard
- Design a set of theoretical tools and methods to describe and evaluate CDN and CDNI networks
- Verify all the designs via experiments in the area of interconnection of multiple CDN networks

Based on the analysis of current state of standardization in this field, it was concluded that the best possible way to disseminate the work is via contributions to the ETSI CDNI standard. These contributions, when combined together, will form a fully functional CDNI standard architecture.

ETSI is a standardization organization that took a complementary role to IETF. While IETF works to define the basics of CDN Interconnection, ETSI directs its effort towards the research of various additional functionalities and more specific use cases that CDN interconnection can offer.

Because ETSI is a much smaller group working a lot closer together than IETF, its work progress is likely to be much faster. This means that the ETSI standard is likely to be published far ahead of the IETF standard. This means that it will have to include a definition of the basic feature set of a CDN-I protocol. This is a disadvantage from practical point of view (the basic feature set is not likely to ever be directly implemented in the industry). However, from the point of view of impact creation, this is an amazing possibility. While the fact that this standard is scheduled to be released too early makes it practically unusable from the industry‘s point of view, it is undisputable that any and all ideas published as part of the standard will be used as a template for the IETF standard that will most likely be adopted by the industry.

As such this group is the perfect breeding ground for innovative ideas in this field and is believed to be the best possible ground for disseminating CDN-I at this moment.

Another goal of the thesis is to design a set of theoretical tools and methods to be used for evaluation of the standard once it is implemented in various production environments. These tools are meant to be used by the actual players in the industry to evaluate the impact of CDNI in their own environments.

To verify the functionality of the architecture and the procedures developed as a part of the thesis, various experiments in the area of interconnection of multiple CDN networks will also executed.

## 4. CDNI Protocol Design

### 4.1 Basic Principles of the CDNI Protocol

Content Delivery Networks (CDN) are systems for the efficient delivery of digital objects (e.g. files with multimedia content as video on demand or other file types) and multimedia streams (e.g. live television streams) over IP networks to many end points and viewers. Typically, a CDN consists of one or more servers that deliver the digital objects and/or streams, and a management/control system. The management/control system takes care of content distribution, request routing, reporting, metadata and other aspects that make the system work.

Currently implemented CDNs are based on a widespread network of distribution nodes controlled by a functional entity taking care of content distribution decisions. The control functional entity keeps track of all content locations, manages distribution amongst the distribution nodes and/or clusters and also decides which distribution node should serve a client request.

Every CDN may have a specific network footprint in which it can deliver content effectively. This footprint depends on the locations of distribution nodes that the CDN controls. It is often inefficient to build new distribution nodes that are not in the region in which the company running the CDN is operating. A solution is to interconnect two CDNs and share existing infrastructure for content delivery. Interconnection of CDNs is achieved by interconnecting the centralized functional entities that represent the logic behind the decisions in each CDN. This is called CDN-I or Content Delivery Network Interconnection.

The CDNI standard defines the architecture and principles of the interconnection between CDNs according to the requirements defined in ETSI TS 102 990 [17], which contains informational use cases for CDN interconnections.

### 4.2 Functional Architecture for CDN Interconnection

The functional architecture is based on a multi-layer architecture that enables separate functionalities used for interconnection of CDNs that may involve up to 3 different level of functionalities and related reference points required. The first layer is responsible for content distribution, the second for controlling of content and request, and the third for the controlling of the interconnection itself. The CDN-I functional architecture should enable CDNs to agree on a minimal level of interconnection, depending on available capabilities and their needs. In case of CDN peering, both interconnected CDNs may be in the role of upstream and/or downstream CDN, depending of direction of content request.

A CDN Interconnection Control Function (ICF) shall manage, create, terminate and exchange CDN networks properties, status report required for CDN interconnection between two or more CDNs (CDN peers).

The Request and Content Control Function (RCF) is responsible for content control and request routing as well as exchanging metadata related to content control. [12] [13]

Distribution of Content Function (DCF) is responsible for distribution of content between CDNs in form of files, streams, and metadata.

### 4.3 CDNI Protocol Procedures

The basic process of CDN interconnection may consist of three basic phases:

- Interconnection establishment, during which the CDNs negotiate the interconnection.

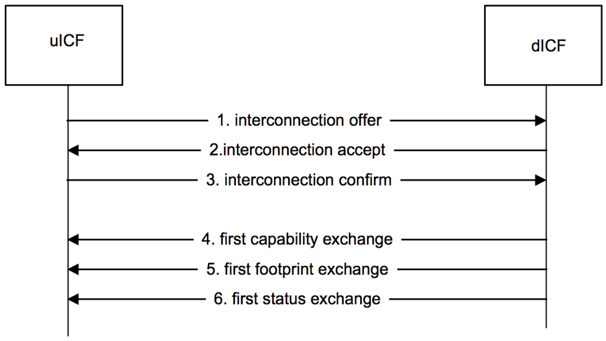


**Figure 1: Interconnection establishment procedure.**

- Interconnection phase, during which the CDNs are fully interconnected and able to share their resources.
- Interconnection release, during which the interconnection between the CDNs is released.

The interconnection establishment is a procedure in which an uCDN and a dCDN begin with no relationship between each other and proceed by exchanging all the information needed to verify each other's identity and thus establish a secure communication channel between each other. This communication is between the ICF's of the CDN's and spans the first three steps of the procedure. The rest of the phase consists of three sub procedures in which the dCDN informs the uCDN about its capabilities, footprint and starts notifying it about its status. After the uCDN receives the first positive status exchange from the dCDN, it can safely assume that the dCDN is ready to process its requests. The conclusion of this phase is the establishment of communication between dRCF and uRCF and the whole CDNI relationship moves to its interconnection phase (Figure 1).

The second phase is simply called interconnection phase. In interconnection phase shall be both CDNs already fully interconnected and through their interconnection interfaces ICF's and RCF's. Interconnection interfaces are fully useable by both CDNs and agreed capabilities accessible over CDN interconnection. The general procedure of the interconnection phase consists of concurrent executions of multiple separate procedures (Figure 2).

The interconnection release is a phase that begins when an uCDN or a dCDN decides to release a CDN peering relationship with each other (for a specific ICF-ICF relation, and associated RCF-RCF relations). This means that the procedure has two variants, depending on whether it was initiated by the uCDN or the dCDN. In both cases it concludes the interconnection relationship by cleanly finishing all outstanding actions between the two CDN's (Figure 3).

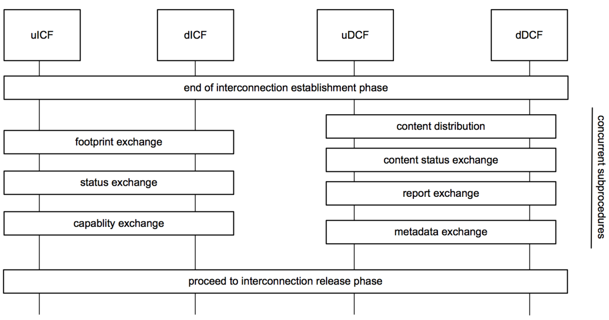


**Figure 2: Interconnection phase.**

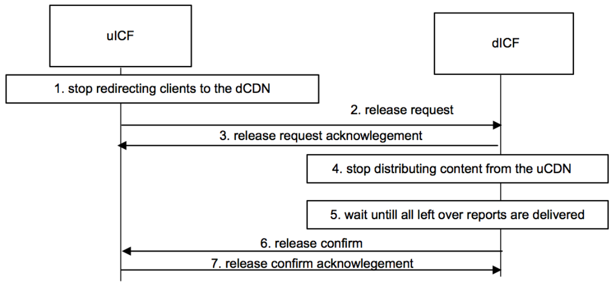


**Figure 3: Interconnection release phase.**

## 5. Verification of Proposed Designs

This chapter describes information about all the experiments executed in relation to this thesis.

From practical point of view, the two most important products of such nature are the CDN laboratory and an experimental CDN-I adapter software that can be used to interconnect multiple CDN's together. It was also necessary to set up a virtualization laboratory in order to be able to fully test all the elements of the environment.

The base of all the interconnection experiments is a set of infrastructural components that form a fully functional laboratory for testing interconnected Content Delivery Networks.

The laboratory is comprised of two separate parts. One of them is a fully functional CDN based on Cisco CDS technology and the other one is a virtualization platform for simulating access networks and clients. The virtualization platform is implemented using Linux, Xen and VMware. The whole laboratory is interconnected by the means of a switched gigabit Ethernet network.

The high level overview of the laboratory is visualized in Figure 4. The diagram also shows which entities are internal and which are external to the laboratory. Only the internal entities are fully controllable.

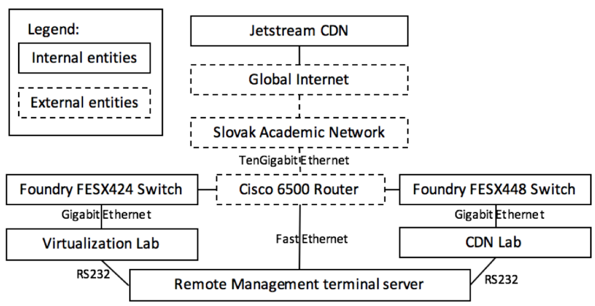


**Figure 4: CDNI laboratory overview.**

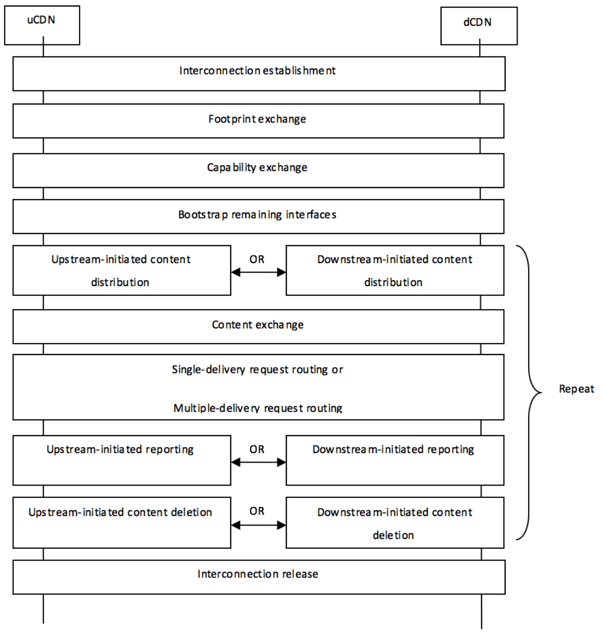


**Figure 5: Basic CDN interconnection life cycle.**

## 5.1 CDNI Adapter

The CDN-I adapter is a piece of software whose main goal is to make it possible to enhance any CDN with ETSI-compliant CDN interconnection capability. This can be achieved by exploiting the APIs already present in most CDN solutions.

The CDN-I adapter is capable to demonstrate complete basic functionality according to the ETSI standard. This means that it is able to execute all the mandatory procedures referenced by the basic interconnection lifecycle scenario of the standard. The basic overview of this scenario is visualized in Figure 5.

The CDNI adapter is a PHP based application with MySQL storage backend. The code of the application is logically separated into uCDN and dCDN functionality. It is also written in a way that will make it very simple to modify to adapt to any CDN other than those already supported. Support for a dummy CDN, Cisco CDS and Jetstream CDN are included. The procedures that are already implemented are:

- Interconnection establishment
- Footprint exchange
- Capability exchange
- Bootstrapping of remaining interfaces
- Single-delivery request routing
- Interconnection release

These procedures make it possible to establish basic interconnection between two CDN‘s from different vendors. In fact the first successful pairing of two CDN‘s using this application was globally the first ever standard-based interconnection of two CDN‘s.

## 5.2 CDNI Adapter Testing Overview

Four different CDN networks were used for evaluation of the functional capabilities of the CDNI adapter (and the underlying CDNI protocol).

- The first one is a CDN network based on Cisco VDS-IS located in our laboratory in Bratislava.
- The second one is a CDN network based on Cisco VDS-IS located in Delft, The Netherlands, in the laboratory of the TNO research institute.
- The third one is a CDN network operated by Jetstream (a commercial CDN provider in the Netherlands).
- The last one is a dummy CDN with a single node (apache http server) located our laboratory in Bratislava.

We used the API‘s provided by the various solution to map them to the procedures specified in this thesis (compliant with the ETSI CDNI standard).

The adapter module implementation for the Cisco-based CDN was straightforward because all the required API calls were supported out of the box and the documentation was on an excellent level.

For the Jetstream Streamzilla CDN the implementation was a bit harder because the CDN does not support on demand ingestion of content (it only supports pre-positioning). We had to use an intermediary server that transparently uploads content items to the data store of the CDN as necessary.

The dummy CDN was developed only as a testing platform for this thesis and as such was designed from the beginning to be interoperable with the CDN adapter. There was obviously no problem integrating with it as we had full control of all its code.

## 5.3 CDNI Adapter Test Results

All the procedures of the CDN‘s were tested with each other to see if the functionality was successfully implemented. Keep in mind that ALL the inter-CDN messaging is done purely by the means of the procedures compliant with the ETSI CDNI standard. There is no inter CDN communication of any other kind. Communication between the adapter and specific CDNs is on the other hand always specific to the particular CDN (with the exception of the two Cisco-based CDNs that share the same API, they differ only in their location).

| | Cisco VDS-IS Bratislava uCDN | Cisco VDS-IS Delft uCDN | Jetstream StreamZilla uCDN | Dummy CDN uCDN |
|---|---|---|---|---|
| Cisco VDS-IS Bratislava dCDN | | Success | Success | Success |
| Cisco VDS-IS Delft dCDN | Success | | Success | Success |
| Jetstream StreamZilla dCDN | Success | Success | | Success |
| Dummy CDN uCDN | Success | Success | Success | |

**Figure 6: Interconnection establishment / Footprint exchange / Bootstrapping of remaining interfaces / Interconnection release - test results.**

| | Cisco VDS-IS Bratislava uCDN | Cisco VDS-IS Delft uCDN | Jetstream StreamZilla uCDN | Dummy CDN uCDN |
|---|---|---|---|---|
| Cisco VDS-IS Bratislava dCDN | | Success | Success | Success* |
| Cisco VDS-IS Delft dCDN | Success | | Success | Success* |
| Jetstream StreamZilla dCDN | Success | Success | | Success* |
| Dummy CDN uCDN | Success* | Success* | Success* | |

**Figure 7: Capability exchange - test results.**

| | Cisco VDS-IS Bratislava uCDN | Cisco VDS-IS Delft uCDN | Jetstream StreamZilla uCDN | Dummy CDN uCDN |
|---|---|---|---|---|
| Cisco VDS-IS Bratislava dCDN | | Success | Failed** | Success |
| Cisco VDS-IS Delft dCDN | Success | | Failed** | Success |
| Jetstream StreamZilla dCDN | Partial Success* | Partial Success* | | Partial Success |
| Dummy CDN uCDN | Success | Success | Failed** | |

**Figure 8: Single-delivery request routing test results.**

*The code had to be modified to resolve overlap of footprint. This was just an issue of the actual implementation, as the standard does not dictate how to handle overlapping footprints. This decision is left to be decided by specific CDN algorithms. The result in our case was that we were not able to pick which CDN to use for an overlapping footprint. Our code makes the decision based on longest prefix priority.

*Works only for pre-positioned content because of Jetstream CDN limitations

**We were not able to control the routing system of the JetStream CDN. Implementing this feature would require creation of an additional API for routing in the Jetstream CDN.

## 6. Conclusion

Internet is without a doubt the most important technology for communicating information across the globe today. As such it is crucial to make sure that the infrastructure behind it can scale to support any current and future needs.

One of the main obstacles in this area is the fact that point to multipoint communication was never successfully resolved on a network level. In fact the most successful method of addressing this issue nowadays is by addressing it on application level, via Content Delivery Networks. They are definitely a technology that can help scale mass distributed content for many years to come.

Today‘s Content Delivery Networks have one significant drawback which is rooted in the fact that majority of them are centralized and managed by a single entity. This is in fact the opposite of the values of de-centralization, neutrality and freedom of speech that made Internet itself such a success. I strongly believe that the way forward is to create a decentralized federation of independent Content Delivery Networks, similar to the concept of independent autonomous system known in today‘s Internet.

### 6.1 Original Contributions

Unlike in today‘s Internet, CDNs do not have a protocol similar to BGP that could be used to link autonomous CDN networks together. The goal of this work was to design such a protocol and prove that it is a viable enabler for the emergence of a global CDN Federation.

The effort was successful. The protocol was not only designed and tested in the environment of our own experimental CDN federation, but majority of its ideas have been accepted as parts of a CDN Interconnection standard published by the European Telecommunication Standards Institute (ETSI TS 182 032).

The next step needs to be taken by the operators and technology vendors themselves. To make their task easier, a method of modeling their Content Delivery Networks and CDN Federations was designed as a part of this work. To verify the viability of this method, it was used to model a sample federation of two CDN networks. The method has proven to be very effective at pinpointing specific areas of possible optimizations in these networks.

**Acknowledgements**. This work is a partial result of the Research and Development Operational Program for the projects Support of Center of Excellence for Smart Technologies, Systems and Services, ITMS 26240120005 and Research and Development Operational Program for the projects Support of Center of Excellence for Smart Technologies, Systems and Services II, ITMS 26240120029, co-funded by ERDF. It is also a part of APVV-15-0731 project and VEGA 1/0836/16.

## References

[1] Cisco Systems, Inc., Cisco Visual Networking Index: Forecast and Methodology, 2010-2015 *Report, 2012*

[2] N. Bartolini, E. Casalicchio, and S. Tucci., A Walk Through Content Delivery Networks *In Proceedings of MASCOTS 2003, LNCS Vol. 2965/2004, pp. 1-25, April 2004.*

[3] J. Dilley, B. Maggs, J. Parikh, H. Prokop, R. Sitaraman, and B. Weihl., Globally Distributed Content Delivery *IEEE Internet Computing, pp. 50-58, September/October 2002*

[4] A. Vakali, and G. Pallis, Content Delivery Networks: Status and Trends *IEEE Internet Computing, IEEE Computer Society, pp. 68-74, November-December 2003*

[5] A.M.K. Pathan, and R. Buyya A Taxonomy and Survey of Content Delivery Networks *Grid Computing and Distributed Systems Laboratory, The University of Melbourne, Australia. February, 2007*

[6] G. Pallis, and A. Vakali, Insight and Perspectives for Content Delivery Networks *Communications of the ACM, Vol. 49, No. 1, ACM Press, NY, USA, pp. 101-106, January 2006*

[7] J. Postel, User Datagram Protocol, *Internet Engineering Task Force RFC 768, August 1980*

[8] J. Postel, Transmission Control Protocol *, RFC793, IETF Internet Standard, 1981.*

[9] R. Buyya, A. M. K. Pathan, J. Broberg, and Z. Tari, A Case for Peering of Content Delivery Networks, *IEEE Distributed Systems Online, Vol. 7, No. 10, IEEE CS Press, Los Alamitos, CA, USA, October 2006.*

[10] K. L. Johnson, J. F. Carr, M. S. Day, and M. F. Kaashoek, The Measured Performance of Content Distribution Networks, *Computer Communications, Vol. 24, No. 2, pp. 202-206, February 2001.*

[11] O. Ardaiz, F. Freitag, and L. Navarro, Improving the Service Time of Web Clients Using Server Redirection, *ACM SIGMETRICS Performance Evaluation Review, Vol. 29, No. 2, ACM Press, NY, USA, pp. 39-44, 2001.*

[12] M. Szymaniak, G. Pierre, and M. van Steen, Netairt: A DNS-based Redirection System for Apache, *In Proceedings of International Conference WWW/Internet, Algrave, Portugal, 2003.*

[13] L. Wang, V. S. Pai, and L. Peterson, The Effectiveness of Request Redirection on CDN Robustness, *In Proceedings of 5th Symposium on Operating Systems Design and Implementation, Boston, MA, USA, pp. 345-360, December 2002.*

[14] D. C. Verma Content Distribution Networks: An Engineering Approach *John Wiley and Sons, Inc.,, New York, 2002.*

[15] J. Jung, B. Krishnamurthy, and M. Rabinovich, Flash Crowds and Denial of Service Attacks: Characterization and Implications for CDNs and Web Sites, *In Proceedings of the International World Wide Web Conference, pp. 252- 262, May 2002.*

[16] European Telecommunications Standards Institute, Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); Content Delivery Network (CDN) Architecture *ETSI TS 182 019, September 2011*

[17] European Telecommunications Standards Institute, CDN Interconnection, use cases and requirements *ETSI TS 102 990, March 2012*

[18] European Telecommunications Standards Institute, Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); Content Delivery Network (CDN) Architecture *ETSI TS 182 019, September 2011*

[19] S. Borst, V. Gupta, A. Walid, Distributed Caching Algorithms for Content Distribution Networks *Proceedings of IEEE INFOCOM, 2010.*

[20] B. Cain, S. Deering, I. Kouvelas, Internet Group Management Protocol, Version 3 *Internet Engineering Task Force RFC 3376, October 2002*

## Selected Papers by the Author

A. Binder, O. M. Van Deventer, B. Wissingh, Overview and Results in Tests from CDN Interconnection Standard published by ETSIPresented on *CDN World Summit, London, 1st-3rd october 2013*

O. M. Van Deventer, B. WissinghM, A. Binder, Standards and developments on CDN federationsPresented on *CDN World Forum, London, 27-6-2013*

A. Binder, I. Kotuliak, Content delivery network interconnect: practical experience. In *ICETA 2013: 11th IEEE International Conference on Emerging eLearning Technologies and Applications. October 24-25, 2013 Stary Smokovec, Slovakia. [S. l.] : IEEE, 2013, s.CD-ROM, s. 29-33. ISBN 978-1-4799-2161-4.*

A. Binder, T. Boros, I. Kotuliak, A SDN Based Method of TCP Connection Handover. In *Information and communication technology : Third IFIP TC 5/8 International Conference, ICT-EurAsia 2015, and 9th IFIP WG 8.9 Working Conference, CONFENIS 2015, Held as Part of WCC 2015, Daejeon, Korea, October 4-7, 2015, proceedings. 1. vyd. Cham : Springer, 2015, S. 13-19. ISBN 978-3-319-24314-6. V databaÌĄze: WOS: 000366189100002 ; SCOPUS: 2-s2.0-84951170956.*

R. Rostecky, I. Kotuliak, A. Binder, Practical experience with CDN interconnection, Presented on *2016 International Conference on Systems, Signals and Image Processing (IWSSIP), Bratislava, 2016 , pp. 1-5. doi: 10.1109/IWSSIP.2016.7502748*