

Access Control in BitTorrent P2P Networks Using the Enhanced Closed Swarms Protocol

Vladimir Jovanovikj, Dušan Gabrijelčič, Tomaž Klobučar

Laboratory for Open Systems and Networks

Jožef Stefan Institute

Ljubljana, Slovenia

E-mail: vladimir@e5.ijs.si

Abstract— The future content delivery platforms are predicted to be efficient, user-centric, low-cost and participatory systems, with social and collaborative connotation. The peer-to-peer (P2P) architectures, especially ones based on BitTorrent protocol, give a solid basis for provision of such future systems. However, current BitTorrent P2P networks lack flexible access control mechanisms. In this paper enhancements to existing access control mechanism for BitTorrent systems – the Closed Swarms protocol are presented, providing additional flexibility in access control mechanism, enabling fine grained security policies specification and enforcement. The enhancements fulfill a number of content providers' requirements and promise efficient, flexible and secure content delivery in future content delivery scenarios.

Keywords— access control, P2P, BitTorrent, flexible policy, Closed Swarms

I. INTRODUCTION

It is envisaged that in the future people will consume 3D content enriched with additional media types and technologies that will engage more of our senses and will provide us immersive experience. People will have the ability to create virtual and personalized environments that will correctly simulate the real world and could have a variety of everyday applications. The virtual environments together with enriched 3D content will bring the communication between people to a higher level, and at the same time will enhance the users' entertainment. Moreover, they will foster human creativity even more and the current trend of people to be not only consumers but also producers of media content is expected to grow [1].

Future content delivery platforms will need to be able to provide efficient delivery of such high quality media content (streaming and stored), on-demand or live to the consumers, with an excellent quality of service. They are predicted to be user-centric and capable of considering the social aspects of the users, as well as the data being delivered. In order to become economically successful, the future content delivery platforms will have to be suitable for large and small size content providers and to be low-cost. This can be achieved if they are participatory and collaborative systems, in which all customers will become actively involved in the content delivery process. Because of its characteristics the peer-to-peer (P2P) architectures gives a solid basis for future provision of such systems. Indeed, one of its most prominent representative [2] – the BitTorrent protocol [3] has already

proved to be scalable, robust and efficient in delivery of large audio and video data, and suitable for live streaming and social interaction between its users [4][5][6]. Thus, BitTorrent promises to be a suitable P2P protocol for future P2P-based content delivery platforms.

In short, with BitTorrent peers exchange small and fixed size pieces of the content file. A group of peers sharing the same file is called a swarm. A peer needs to acquire a so called torrent file in order to start downloading. The torrent file contains the needed information for the protocol initiation. The sharing process is coordinated either by a central server – the tracker or by the participants themselves – using the DHT [7] protocol. BitTorrent uses tit-for-tat policies to provide fairness in the delivery process [8]. Peers that continue to upload after they have downloaded the whole content file (seeds) improve the downloading process of the other peers (leeches).

The future P2P-based content delivery platforms need to be secure and trusted in order to be widely accepted and used. The importance of security as well as the main security requirements for P2P networks have already been emphasized in [9][10]. Among them access control is considered basic and standard, especially by content providers. The access control in the P2P-based content delivery systems is quite difficult to accomplish because of the basic properties of the system: 1) the content consumers are directly involved in the process of content distribution, i.e. peers exchange the content among themselves; and 2) the system tends towards full decentralization, without even a single central party for administration.

The main goal of this paper is to propose several enhancements of an existing access control mechanism for BitTorrent P2P networks – the Closed Swarms protocol [11], that we believe will provide a flexible access control mechanism for future P2P-based content delivery platforms applicable in various scenarios. First, we give an overview of the existing approaches for access control in BitTorrent P2P networks in Section II. Then, we describe the motivation for enhancing the Closed Swarms protocol in Section III. We present our proposed enhancements in Section IV and furthermore discuss them in Section V. Finally, we conclude the paper and present our future work in Section VI.

II. RELATED WORK

Access control in P2P content delivery systems can be achieved either directly protecting the content being delivered or controlling the content delivery process.

An access control mechanism directly protecting the content is proposed by Zhang et al. [12]. It is basically a digital rights management (DRM) mechanism for BitTorrent, based on using trusted tracker and initial seed, as well as using trusted content viewer on the client side. The main idea behind their schema is existence of a single plaintext copy of the content being delivered, the one at the trusted initial seed, while all the other copies of the content, resting at the peers being part of the content delivery system are uniquely encrypted for every peer, piece by piece. The peers consume the content only with a trusted content viewer, which is responsible for decrypting the content according to the purchased license from the content provider. This scheme is highly dependable on the tracker, which is far from full decentralization and is an obvious security risk – a single point of failure. Moreover, it doesn't provide means for applying flexible content usage policies, even though it is possible to define copyright related usage policies into the license. All this makes this scheme not appropriate for the future P2P-based content delivery platforms. In addition, the encryption and increased communication with the tracker certainly have impact on the performance of the content delivery. It is worth mentioning that providing copyright protection in a fully decentralized environment that favours open source software is a task very difficult to fulfill.

Another mechanism for direct protection of the content is described by Jimenez et al. [13]. In their scheme, the provider first encrypts the content before it is being distributed among the peers. Only peers that commit a payment and satisfy the provider's policies are authorized to receive the decryption key and consequently are able to decrypt the content. Although this mechanism is capable for implementing a certain access control policies (for example based on geolocation), it depends only on one cryptographic key, which makes it to be easily compromised.

Private tracker [14] extension of the BitTorrent protocol is an access control mechanism for controlling the delivery process. It restricts access in the system by simply not giving information about the participants to unauthorized users, i.e. users that do not meet a certain criteria, such as minimum upload-to-download ratio. This mechanism is not appropriate for future P2P-based content delivery platforms as it is highly centralized. Also, it depends on peers using only one private tracker at a time as a peer discovery mechanism, which makes it be easily subverted.

Closed swarms (CS) protocol [11] is an access control mechanism for controlling the delivery process that acts on peer level. It enables peers to recognize the authorized peers and to avoid communication with the non-authorized ones. The distinction between authorized and non-authorized peers in the swarm is made based on possession of an authorization credential called proof-of-access (PoA). The peers exchange their credentials right after they establish connection, in a challenge-response messages exchange. In

most severe case, with the CS protocol only the authorized peers receive service (content). Nevertheless, it is possible to design a system in which both users would receive service (content), but graded – the authorized users would receive additional or better service than the non-authorized ones, for example access to high speed seeds for better performance. The CS protocol can provide access control in an innovative business content delivery system, but only under the same conditions for all authorized users. Moreover, this protocol is vulnerable to man-in-the-middle attacks.

Another access control mechanism for controlling the delivery process that acts on peer level is Lockr [15]. It is a privacy preserving access control mechanism for social networks in general. It is also applicable in BitTorrent P2P networks, for people to control the delivery of their personal content via them. The content owner issues digitally signed social attestations to all persons it has a social interaction with. A social attestation certifies the social relationship between two persons. In order to start exchanging pieces of the content, two peers need to verify their attestations during a social handshake, a form of zero-knowledge protocol. This access control mechanism is fine example of improved privacy in content delivery and in social networks in general. However, it still lacks support of flexible access control policies for the future P2P-based content delivery platforms.

III. MOTIVATION

To motivate our work we describe the following scenario. An international TV broadcaster (a content provider) wants to distribute live TV program to its clients (authorized users) using a P2P-based content delivery platform, based on the BitTorrent protocol. The TV broadcaster aims at achieving fine grained load balancing and optimization of its program delivery process, and restricting its program's availability only in one country (e.g., only in Slovenia) because of the digital rights issues, although it is broadcasting other programs in several countries. Furthermore, the TV broadcaster decides to deliver a service to clients under different conditions. Premium clients, for example, would receive higher content quality (e.g., HD video) for a certain amount of money, whereas basic clients would receive lower content quality (e.g., SD video) for free. This is beneficial from business perspective, as it can increase indirect earnings, and from technical perspective, since it can improve content delivery. Moreover, clients should be able to purchase certain service packages in which they will receive high content quality only during certain time periods, e.g., every day from 18 till 20 hours, during the most popular show. Analysis of the scenario elicited the following requirements.

Requirement 1: Fine grained load balancing and optimization of the delivery process: In BitTorrent live streaming swarm, none of the peers, except the content injector, has the whole content in advance, as seeds in regular swarms do. Instead, seeds in live streaming swarm are special peers with outstanding properties (e.g., high bandwidth), which are always unchoked by the content injector and have the same role in content delivery as the regular seeds – they improve the other peers' download

performance. The seeds are often purposely set by the content provider and behave as a small Content Delivery Network (CDN) [6].

In order to achieve fine grained load balancing and optimization of the delivery process, the content provider (TV broadcaster) can create and maintain a hierarchical structure of seeds in the live streaming swarm, analogical to a hierarchical CDN [16]. This structure is formed by separation of the seeds into layers (levels) according to the priority assigned to them by the content provider (Fig. 1) and placing the seeds at strategic locations. The greater the priority of the seeds a layer contains is – the higher it appears in the structure. The value of the priority defines the level of precedence a seed has among the other peers in the live streaming swarm (seeds and leeches). Normally, the content injector and the seeds establish a connection to any peer in the swarm regardless of its priority, as long as they have a free connection. However, when a lack of free connection occurs, the connections with seeds having lower priorities or with leeches will be terminated in favour of seeds having greater priorities.

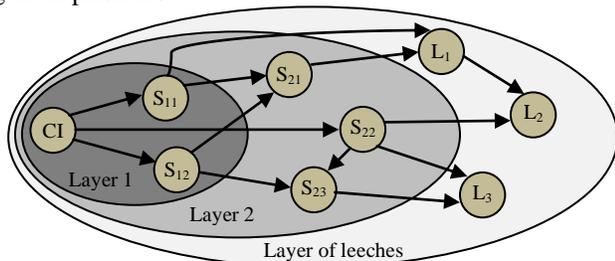


Figure 1. Hierarchical structure of a live streaming swarm: the content injector (CI) is not part of any layer; the seeds (S) from layer 1 have a priority (e.g., 20) greater than the seeds from layer 2 (e.g., 10); the leeches (L) are all placed in one layer and do not have any priority.

Two mechanisms are needed for the process of creation and maintenance of a hierarchical structure of seeds in a live streaming swarm.

Sub-requirement 1.1: Automatic introduction of a seed: Seeds download content only from the content injector or other seeds, which explicitly know them by maintaining lists of their identifiers (e.g., IP address and port number). However, these lists are maintained manually – something that becomes impractical for a large swarm (like in the scenario above) and very difficult for creation and maintenance of a hierarchical layered structure. Therefore, a mechanism for automatic introduction of a seed in the live streaming swarm is needed, that will also place the seed in a specific layer of the hierarchical structure.

Sub-requirement 1.2: Suitable peer discovery: This mechanism is needed to enable quick transport of the content from the content injector towards the lowest level of the structure of seeds, and consequently to the regular peers (clients). Currently, none of the peer discovery mechanisms the BitTorrent protocol supports (e.g., the tracker [3] or the DHT [7] protocol), takes into consideration a hierarchical structure of a live streaming swarm.

Requirement 2: Restriction of the content delivery based on peer location: According to this requirement, only peers

inside one country are allowed to receive an authorization credential and join the swarm. The physical location of a peer on country level can be determined by using the Internet geolocation technology. Although tactics for evasion of this technology do exist, it is considered sufficient for compliance with the legal regulations [17]. The CS protocol needs to be properly extended in order to take into consideration the output of the Internet geolocation technology in the access control decision.

Requirement 3: Provision of different content quality in the same swarm: The content provider needs to create only one content stream by using a scalable video coding technique, but encoded in several layers [18]. Then, by specifying in the authorization credentials which layers the holders are allowed to receive, peers can easily determine which content pieces should provide to them. For example, premium clients would be authorized to receive content pieces from all the encoding layers, while other clients – only from fewer layers.

Requirement 4: Temporal constraints: In addition to the previous requirement, the authorization credential can also specify temporal constraints, for example, when the allowed content layers would be provided to the clients. This can even provide a basis for business models in the content delivery process by creating different service packages for the clients.

IV. THE ENHANCED CLOSED SWARMS PROTOCOL

We believe that after enhancement, the Closed Swarms protocol fulfills the requirements from Section III and becomes resistant to man-in-the-middle attacks. Before presenting the proposed enhancements of the CS protocol, in short we describe the format of the authorization credential and the message exchange process in the CS protocol, explained in detail in [11].

The authorization credential (called Proof of Access) of an arbitrary peer A (1) contains information about: the specific swarm – its identifier (SwarmID) and public key (K_S); the credential holder, defined by its public key (K_A); and the expiry time of the credential (ET). The credential issuer, usually the content provider in correlation with a payment system¹, digitally signs this information with the private key of the swarm (K_S^{-1}). The authorization credential is valid only when all the fields and the digital signature are correct.

$$SwarmID, K_S, K_A, ET, \{SwarmID, K_S, K_A, ET\}_{K_S^{-1}} \quad (1)$$

Two peers, an initiator – peer A, and a swarm member – peer B, exchange their credentials in a challenge-response message exchange process:

$$A \rightarrow B: SwarmID, N_A \quad (2)$$

$$A \leftarrow B: SwarmID, N_B \quad (3)$$

$$A \rightarrow B: PoA_A, \{N_A, N_B, PoA_A\}_{K_A^{-1}} \quad (4)$$

$$A \leftarrow B: PoA_B, \{N_A, N_B, PoA_B\}_{K_B^{-1}} \quad (5)$$

¹ The credential issuer signs credential for all swarms it is responsible for, by using their private keys. Although there is no specific protocol of issuing the credentials, the process is explained in detail in [11].

First, with (2) and (3) they exchange the identifier of the swarm they want to join/are part of and randomly generated nonces (N_A/N_B). Then, with (4) and (5) they exchange their credentials (PoA_A/PoA_B) followed by a concatenation of the previously exchanged nonces and the credential, digital signed with their private keys (K_A^{-1}/K_B^{-1}).

The requirements from Section III can be satisfied by using an access control based on flexible authorization framework and proper policy enforcement. A number of distributed frameworks have already been proposed in the past [19]. Here, we aim at integrating such distributed authorization framework in the CS protocol. Furthermore, although the protocol uses authorization credentials containing public key for owner identification, random nonces for message freshness and digital signatures for message authentication, it still remains vulnerable to man-in-the-middle attacks. An attacker can interfere in the communication between two authorized peers by simply relaying the messages between them. After the authorized peers successfully finish the protocol and start the content delivery, the attacker will be able to read all the exchanged content pieces, since they are not encrypted. We propose encryption of the content pieces with a shared secret key as a countermeasure for this attack.

The format of the extended authorization credential is as follows:

$$\begin{aligned} & SwarmID, K_S, K_A, ET, Rules_A, \\ & \{SwarmID, K_S, K_A, ET, Rules_A\}_{K_S^{-1}} \end{aligned} \quad (6)$$

The newly introduced field – Rules contains conditions under which the credential holder is authorized by the credential issuer to join the swarm and receive the requested service (e.g., content quality, level of prioritized treatment). The format of this field, described with the ABNF notation [20], is given below:

$$Rules = [General] [Per-piece] \quad (7)$$

$$General = conditions \quad (8)$$

$$Per-piece = conditions \quad (9)$$

$$\begin{aligned} conditions &= condition [log-operator conditions] / \\ & "(" conditions ")" \end{aligned} \quad (10)$$

$$log-operator = "and" / "or" \quad (11)$$

$$\begin{aligned} condition &= variable operator value / \\ & variable operator variable \end{aligned} \quad (12)$$

$$operator = "=" / "!=" / "<" / "<=" / ">" / ">=" \quad (13)$$

$$variable = 1ALPHA *99(ALPHA / DIGIT) \quad (14)$$

$$\begin{aligned} value &= 1*10DIGIT / 1*10DIGIT "." DIGIT / \\ & "" 1*10ALPHA "" \end{aligned} \quad (15)$$

The Rules field contains two groups of conditions: a general group and per-piece group. The former contains conditions evaluated every time the credential holder connects to another peer, as well as at specific time (in case of time conditions), whereas the latter contains conditions evaluated on every piece request from the credential holder. In each condition a value of a specific environment variable is compared to other variable or a predefined value. The values of the environment variables are dynamically assigned from the environment of the evaluating peer or from another field, as described later. Each group of conditions is positively

evaluated only if the compound logical sentence produces a truth value.

Having on mind the roles of peers A and B, the extended and modified message exchange process goes as follows:

$$A \rightarrow B: Version_A, SwarmID, N_A \quad (16)$$

$$A \leftarrow B: Version_B, SwarmID, N_B \quad (17)$$

$$\begin{aligned} & A \rightarrow B: PoA_A, ReqService_A, \\ & \{N_A, N_B, PoA_A, ReqService_A\}_{K_A^{-1}} \end{aligned} \quad (18)$$

$$\begin{aligned} & A \leftarrow B: PoA_B, Info_B, Peers_B, \{K_{AB}\}_{K_A}, \\ & \{N_A, N_B, PoA_B, Info_B, Peers_B, \{K_{AB}\}_{K_A}\}_{K_B^{-1}} \end{aligned} \quad (19)$$

$$A \leftarrow B: Info_B, \{N_A, N_B, Info_B\}_{K_B^{-1}} \quad (20)$$

First, the peers exchange the latest version of the protocol they support (Version), together with the swarm identifier and the randomly generated nonce, with (16) and (17). Then, peer A sends its authorization credential and specifies the service properties (ReqService) it wants to receive with (18). Next, peer B evaluates the service request. If it is according to peer A's authorizations and if peer B can provide the requested service (for example it has an available connection – a free or one to a peer with lower priority that can be terminated), it will enable upload to peer A. Otherwise, upload will be disabled. In both cases, it will first send (19) in order to clarify the process outcome (Info) and to recommend other swarm members for contacting (Peers) to peer A. In positive case (19) will also contain a symmetric key (K_{AB}), generated by peer B and encrypted with peer A's public key, which will be used for encryption of the provided service – the content pieces. On the other hand, in negative case this field will be empty. After a positive (19), peer B starts to upload content to peer A. It also continues to verify the validity of the peer A's credential and to evaluate every piece request according to its authorizations. When a violation occurs, it will send (20) as notification and it will stop uploading. In addition, the protocol will be aborted when one of the peers sends an invalid credential, an incorrect digital signature or a different swarm identifier.

The positive outcome of the message exchange process is one way upload, from peer B to peer A. If peer B is also interested in downloading content from peer A while uploading, it needs to start the same exchange process, but now acting as an initiator.

The formats of the newly introduced fields are as follows. First, the Version field is two bytes and states the protocol version. For example, 02_{HEX} denotes the enhanced CS protocol version. Since this or any future protocol extension or modification results in a new version, peers need to be aware of the versions they support in order to have successful communication. In this way, means for backward compatibility between CS protocol versions can be possible. Next, the description of the ReqService field format, by using the ABNF notation, is:

$$ReqService = ["(" assignment ")"] ["," ReqService] \quad (21)$$

$$assignment = variable "," value \quad (22)$$

$$variable = 1ALPHA *99(ALPHA / DIGIT) \quad (23)$$

$$\begin{aligned} value &= 1*10DIGIT / 1*10DIGIT "." DIGIT / \\ & "" 1*10ALPHA "" \end{aligned} \quad (24)$$

It contains pairs that actually define an assignment of a certain value to a specific environment variable at the

evaluating peer. These values must be assigned to the environment variables before evaluation of the conditions in the Rules field, since they influence the evaluation of the policies from the Rules field. The ReqService field can contain information such as requested content quality or level of prioritized treatment. Furthermore, the Info field is two bytes and specifies the identifier of predefined information that clarifies the protocol outcome. This information can confirm that the upload is enabled or state the reason why it is disabled. For example 01_{HEX} means unauthorized service properties requested. Finally, the Peers field is a set of maximum 5 pairs, each denoting a swarm member. A pair contains either IP address (IPv4 or IPv6) or DNS name of the member, together with its port number.

In conclusion, the enhanced CS protocol is an access control mechanism that acts on a peer level that enables peers to exchange the authorization credential and requested service properties between them in a secure manner.

V. DISCUSSION

Together with our proposed enhancements, the CS protocol fulfills the requirements from Section III, and becomes resistant to man-in-the-middle attacks.

To begin with, the introduction of Rules and ReqService fields fully satisfies the desired requirements 2-4, as well as the sub-requirement 1.1. The Rules field provides creation of expressive and flexible access control policies. These policies are contained in the authorization credential itself which makes their modification easy and dynamic. The policies can be tailored to several groups of peers in the swarm, distinguishable on the basis of various criteria, such as role in a swarm (seed or leech), priority, location and allowed content quality (number of stream layers), during different time periods. Now, seeds can automatically join the hierarchical swarm only by receiving appropriate authorization credentials. However, every peer must first explicitly request the properties of the service it wants to receive by specifying them in the ReqService field, in order to have its policy evaluated correctly. In this way, together with the help of the notifications in the Info field, they can even negotiate (to some extent) the service properties they want to receive.

The access control diagram of a request for service is illustrated in Figure 2 (based on [21]). The initiator – peer A sends its authorization credential and specifies the service properties it wants to receive to the swarm member – peer B with message (18). The Rules field is passed to the peer B's Access Control Decision Function (ADF), where the embodied policies are evaluated. On the other hand, the values from the ReqService field are assigned to specific environment variables and together with other environment variables are taken into account during evaluation of the policies. The peer B's Access Control Enforcement Function (AEF) grants or denies the access to the requested service according to the evaluation of the specified policies.

An example of information provided to the evaluating peer's ADF, i.e. the contents of the Rules and ReqService

fields sent by a seed, together with needed environment variables, applicable in the described scenario above is given in Figure 3. The Rules field denotes that the seed is authorized by the credential issuer to join the swarm only when it is located in Slovenia and requests high content quality (e.g., HD video) and prioritized treatment appropriate for level 2 seed. According to the explicitly requested service properties in the ReqService field by the seed and the values of the specific environment variables at the evaluating peer, this policy is positively evaluated at the ADF and the seed is granted access to the swarm.

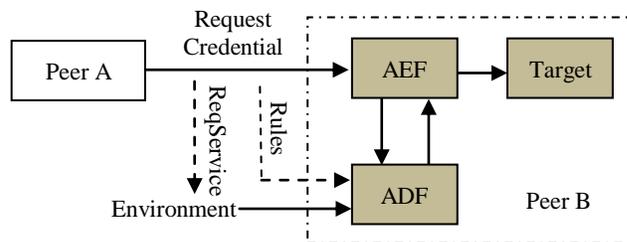


Figure 2. Access control diagram of request for service with the enhanced CS protocol (based on [21]).

Rules:

General:

GEOLOCATION = 'SI' and
 PRIORITY <= 10 and
 CONTENT_QUALITY <= 3

Per piece:

ReqService:

(PRIORITY, 10),
 (CONTENT_QUALITY, 3)

Environment:

GEOLOCATION = 'SI'

Figure 3. Contents of the Rules and ReqService fields sent to a closed swarm member by a level 2 seed (Fig. 1) and the values of the environment variables at the swarm member.

Furthermore, the newly introduced Peers field provides a peer discovery mechanism applicable in hierarchically structured live streaming swarm, which satisfies sub-requirement 1.2 from Section III. The peer discovery mechanism goes as follows. Every peer first contacts the content injector using the CS protocol. If it is authorized to enter the swarm, it will receive by the content injector a list of swarm members from the layer with the highest priority. Then it continues to contact the returned members and to receive information about other members from the swarm, until it creates the number of connections it needs. Peers return information about members from the same layer or the layer with one level lower priority, as long as this priority is greater than or equal to the initiator's priority. This guarantees that peers will always download content only from peers with the same or higher priority in the structure.

In addition, the Version field provides means for backward compatibility. After two peers exchange the

protocol version they support, the peer supporting the higher version can adapt and send appropriate messages to the version the other peer supports. However, this is applicable in specific cases and only to those peers that are not directly concerned with the higher version changes. For example, if the original CS protocol did contain the Version field, the basic clients from the described scenario could use the original protocol, but only when the requirement for restriction of the content delivery based on peer location is not mandatory.

Finally, by encrypting the exchanged content with a secret key we prevent a malicious peer to read it in an unauthorized manner. However, the purpose of the encryption is not to provide confidentiality of the provided service, but only to fight man-in-the-middle attacks. Also, it does not prevent explicit leakage of content to unauthorized peers, which still depends on the good behavior of the authorized peers.

In conclusion, all the desired requirements from Section III can be satisfied with our proposed enhancements, and means for backward compatibility can be achieved. Also, the vulnerability of the protocol to man-in-the-middle attack is fixed.

VI. CONCLUSION AND FUTURE WORK

In this paper we have proposed several enhancements of an existing access control mechanism for BitTorrent P2P networks – the Closed Swarms protocol. The enhancements provide additional flexibility in access control mechanism, enabling fine grained security policies specification and enforcement. The enhancements fulfill a number of content providers' requirements and promise efficient, flexible and secure content delivery in future content delivery scenarios. Our future work includes integration of the proposed enhancements into the P2P-Next delivery platform (<http://p2p-next.org>) and their evaluation.

REFERENCES

- [1] Future Media Internet Task Force: Research on Future Media Internet. A white paper, (2009). Obtained on April 5, 2011 from: ftp://ftp.cordis.europa.eu/pub/fp7/ict/docs/netmedia/research-on-future-media-internet-2009-4072_en.pdf
- [2] Schulze, H., Mochalski, K.: Internet Study 2008/2009 (2009). Obtained on April 5, 2011 from: <http://www.ipoque.com/userfiles/file/ipoque-Internet-Study-08-09.pdf>
- [3] Cohen, B.: BitTorrent protocol specification, (2008). Obtained on April 5, 2011 from: http://www.bittorrent.org/beps/bep_0003.html
- [4] Pouwelse, J. A., Garbacki, P., Wang, J., Bakker, A., Yang, J., Iosup, A., Epema, D. H. J., Reinders, M., van Steen M. R., Sips H. J.: Tribler: A social-based based peer to peer system. 5th Int'l Workshop on Peer-to-Peer Systems (IPTPS), Santa Barbara, USA (2006)
- [5] Pandey, R.R., Patil KK.: Study of BitTorrent based Video on Demand Systems. International Journal of Computer Applications, Vol.1, No.11, pp. 29-33. Foundation of Computer Science, (2010)
- [6] Mol, J.J.D., Bakker, A., Pouwelse, J.A., Epema, D.H.J., Sips, H.J.: The Design and Deployment of a BitTorrent Live Video Streaming Solution. In: 11th IEEE International Symposium on Multimedia, ISM '09, pp. 342-349. IEEE Computer Society, Los Alamitos (2009)
- [7] Loewenstern, A.: DHT Protocol, (2008). Obtained on April 5, 2011 from: http://bittorrent.org/beps/bep_0005.html
- [8] Cohen, B.: Incentives Build Robustness in BitTorrent, In: Proceedings of the 1st Workshop on Economics of Peer-to-Peer Systems, Berkeley, USA, (2003)
- [9] International Telecommunication Union – Telecommunication Standardization Sector (ITU-T): Data networks, open system communications and security, Framework for secure peer-to-peer communications. ITU-T Recommendation X.1161, (2008)
- [10] Gheorghe, G., Lo Cigno, R., Montresor, A.: Security and privacy issues in P2P streaming systems: A survey. Peer-to-Peer Networking and Applications. Springer, New York (2010)
- [11] Borch N.T., Arntzen, I., Mitchell K., Gabrijelčić D.: Access control to BitTorrent swarms using Closed Swarms. In: Proceedings of the 2010 ACM Workshop on Advanced Video Streaming Techniques for Peer-to-Peer Networks and Social Networking, AVSTP2P '10, pp. 25-30. ACM, New York (2010)
- [12] Zhang, X., Liu, D., Chen, S., Zhang, Z., Sandhu, R.: Towards digital rights protection in BitTorrent-like P2P systems. In: Proceedings of the 15th ACM/SPIE Multimedia Computing and Networking, San Jose, USA (2008)
- [13] Jimenez, R., Eriksson, L., Knutsson, B.: P2P-Next: Technical and Legal Challenges. The Sixth Swedish National Computer Networking Workshop and Ninth Scandinavian Workshop on Wireless Adhoc Networks, Uppsala, Sweden (2009)
- [14] Harrison, D.: Private Torrents, (2008). Obtained on April 5, 2011 from: http://bittorrent.org/beps/bep_0027.html
- [15] Tootoonchian, A., Saroiu, S., Ganjali, Y., Wolman, A.: Lockr: better privacy for social networks. In: Proceedings of the 5th international conference on Emerging networking experiments and technologies, CoNEXT '09, pp. 169-180. ACM, New York (2009)
- [16] Andreev, K., Maggs, B.M., Meyerson, A., Sitaraman, R.K.: Designing overlay multicast networks for streaming, In: Proceedings of the fifteenth annual ACM symposium on Parallel algorithms and architectures, SPAA '03, pp. 149-158. ACM, New York (2003)
- [17] Muir, J.A., van Oorschot, P.C.: Internet geolocation and evasion. Technical Report TR-06-05, Carleton University, School of Computer Science (2006)
- [18] Asioli, S., Ramzan, N., Izquierdo, E.: Efficient Scalable Video Streaming over P2P Network. User Centric Media, LNICST, vol. 40, pp. 153-160. Springer, Heidelberg (2010)
- [19] Chapin, P., Skalka, C., Wang, X.S.: Authorization in trust management: Features and foundations. ACM Computing Surveys, Vol. 40, pp.1-48. ACM, New York (2008)
- [20] Crocker, D., Overell, P.: Augmented BNF for Syntax Specifications: ABNF. Request for Comments (RFC) 5234, (2008)
- [21] International Telecommunication Union – Telecommunication Standardization Sector (ITU-T): Data networks, open system communications and security, Information technology – Open systems interconnection – Security frameworks for open systems: Access control framework. ITU-T Recommendation X.812, (1995)