SCADDIC: The Implementation and Performance of a Scalable and Secure Architecture for Digital Content Distribution

Valer BOCAN
Department of Computer Science and Engineering
"Politehnica" University of Timişoara, ROMANIA
Alcatel-Lucent România
E-mail: vbocan@dataman.ro

Vladimir CREȚU
Department of Computer Science and Engineering
"Politehnica" University of Timişoara, ROMANIA
E-mail: Vladimir.Cretu@cs.upt.ro

***Abstract*: This paper describes the performance facts and implementation issues of a previously published scalable and secure architecture for digital content distribution. We demonstrate the viability of the proposed architecture by showing the result of our lab implementation and we have some practical advice for industry implementations.**

***Index Terms* – scalable DRM, digital content, distribution, redistribution, secure broadcast, cryptography, secret splitting, implementation, SCADDIC.**

I. Introduction

With the advent of several wireless technologies and with the speed advances we have witnessed in the past few years plus the ubiquity of portable devices, we are likely to see a paradigm shift in the near future, from the on-schedule broadcast to the on-demand broadcast of content. Associated with digital rights management issues, the broadcast of digital content is a topic that has its own chapter in the encyclopedia of communications.

Technologies like Bluetooth, Wi-Fi, 3G and the like, not to mention the large variety of flash memories that can be read and written by mobile devices, have brought a unique set of challenges to the content distributors. If in the pre-wireless era, people used to share content using physical media such as CDs and DVDs, now people are increasingly able and willing to share content using their portable devices, in a point-to-point manner. This has opened the way to digital content redistribution, a process which generates revenue loss to the content provider.

To preserve the digital content from illegal duplication and unauthorized distribution and to ensure that copyright laws are obeyed, content providers have searched for new ways of implementing secure distribution systems, based on Digital Rights Management (DRM) policies. The greatest challenge posed to such a system is to ensure that these policies are effective even after the customer came into the possession of the digital content, especially when he attempts to redistribute it to another user.

The current approach to the above-mentioned problem is to introduce the so-called *compliant devices*, which, by their design, guarantee to respect the DRM policies associated with the multimedia content they store and play. For example, a compliant device will refuse to share its contents with a *non-compliant device*, or to redistribute the multimedia content to another compliant device should the associated DRM policy forbid it to do so.

This paper is based on an earlier work on a scalable digital content distribution architecture, which relies on the secure broadcast technique for distributing the digital content to several clients simultaneously [1]. Along with the implementation, the architecture has also received a reference name: SCADDIC.

The paper is organized in six sections. In Section II we briefly present the system architecture and the parties involved in the digital content distribution and redistribution. Section III gives a bird's eye view of the implementation and section IV gives some performance figures related to various operations that comprise the content distribution schema. Section V makes a side-by-side comparison of the broadcast and unicast techniques and outlines the advantages and disadvantages of both. The final section draws conclusions regarding scalability of our proposed SCADDIC system.

II. System Architecture in brief

Upon practical implementation of the architecture presented in [1], we made slight changes to the roles and names of certain entities such that they better reflect our vision.

The updated architecture contains the following major parts:

- The authority and content distribution part composed of one *Content Master* and several *Content Providers*;
- The consumer network composed of *Clients* or *Content Proxies*.

Parties involved in the SCADDIC scheme are as follows:

A. Content Providers

Content providers (CPs) are parties which own or share the rights to distribute a particular piece of digital content. A consumer which requests the digital content will issue a request to content master which, in turn will subject the request to the approval of all the CPs which share the distribution rights of the digital content. If all CPs approve

the request, then the content master will encrypt and send the digital content to the client (or clients), otherwise it will deny the request.

A single piece of digital content will be broadcast to the clients who request it in a given time frame. This technique reduces the load on the server by allowing simultaneous servicing of several clients.

B. Content Master

The Content Master (CM) is a party trusted by all content providers (CP) and serves as a central point for handling content requests and storing the content itself. When new content is received for storage (registered with the Content Master) it is first encrypted using a random key, then the key is split into several segment which are in turn distributed to the content providers relevant to the content being processed. Since no content is being stored in clear and the keys are split among several entities, it is safe to assume that the Content Master is not a single point of failure, at least not from the security standpoint.

C. Clients

Clients are compliant devices which receive the rights to play the multimedia content along with the content itself. The content may be received from the Content Master or some of its proxies and may optionally purchase the rights to redistribute it. The requirements which must be fulfilled by such a compliant device are outlined in the Trusted Platform Module (TPM) specifications [2]. Each compliant device is endowed at manufacturing time with a pair of asymmetric keys: *the public key* and *the private key*, used for digital signatures and verifications.

D. Content Proxies

Content proxies differentiate from clients in the way that they are acting as relays between the CM and clients. Any client may become a content proxy if he desires so, by signaling this intent to the CM. Since the multimedia content is encrypted, the proxy will not be able to render it if not addressed to it directly, but is able to forward the content to its legitimate destination.

Using content proxies, the CM can reduce its load significantly, since it would no longer be required to sequentially connect to each client that requested the content.

The CM should constantly monitors proxy activities to make sure that they are in accordance with the applicable DRM policies.

III. IMPLEMENTATION

The SCADDIC implementation was done with the proof of concept in mind, mainly to demonstrate that the concept is viable, feasible and functional. Therefore the net result is a lab prototype which may not be secured against various attacks. The prototype does however preserve the ratio of various tasks so that they are relevant to our performance evaluation.

The project is comprised of three sections:

- Content description
- Content handling
- Cryptographic and math operations

Each part of the project is summarized in the rest of this section.

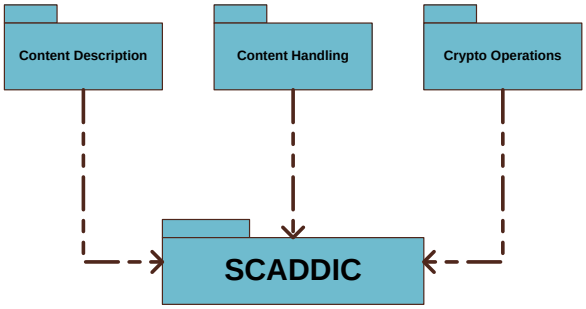

Fig. 1 Bird's eye view on of the SCADDIC system

A. Content Description

Content description deals with classification of the digital content and its metadata. Each piece of content is uniquely identified by a nonce and has associated metadata that describes the rights for rendering it as well as descriptors such as title, author, year of release, bit rate, codec, etc. Also, in order to prevent the content from being tampered with in transit, a checksum mechanism is present.

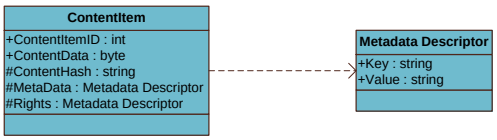

Fig. 2 Content description

B. Content Handling

The content handling package describes the parties involved in the content distribution schema:

- Client – a computer or a mobile device that requests and consumes and redistributes the digital content, compliant with the licenses and rights associated.
- Content Provider – the entity that retains the copyright over the content being distributed. Can work in conjunction with other providers with which it shares the rights.
- Content Master – the trusted party in the SCADDIC schema. Requests that come from the clients are centralized and serviced here, with the approval of the content providers (which may discretionary deny access to a particular client).

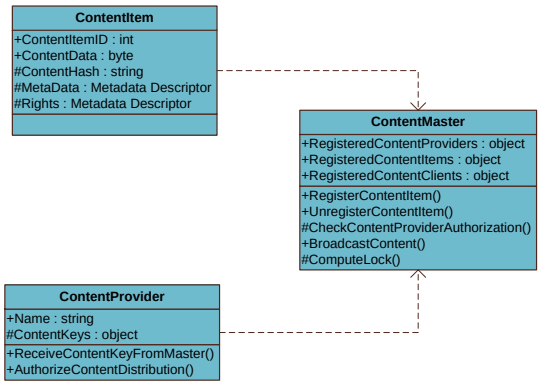

Fig. 3 Content handling

C. *Cryptographic and Math Operations*

In the process of handling the digital content, there are numerous cryptographic and mathematical operations that must be performed. This package is the most complex as it lays the foundation of the entire SCADDIC project.

IV. Performance Facts

The performance of the SCADDIC prototype can be categorized as follows:

- Operations that take place once or at system start-up.
- Operations that are performed upon content distribution/redistribution.

Naturally, we will not be concerned too much about the first category, as the times do not interfere with the normal operation of the system. We will be giving these times anyway, for reference purposes.

The code was compiled using Visual Studio 2008 running on Windows Vista and run on a Lenovo T60 portable computer.

The duration quoted in this section refer to the time needed for the preparation of the digital content (cryptographic operations, hashing, signing, etc.) and does not include the time spent by the content in transit over the transport medium.

A. *Start-up Operations*

Registration of content providers is done when the content master initializes.

Content Provider Count	Registration Time (ms)
1	1
10	10
50	43
100	85
200	169

Table 1. Content provider registration times

The time needed for registration increases linearly with the count of providers. The average registration time is around 0.9 ms.

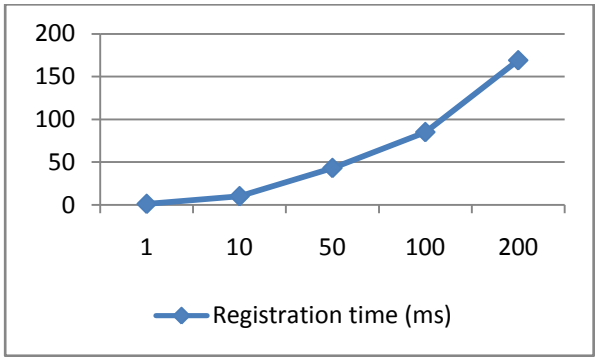

Fig. 4 Content provider registration time variation

Registration of clients would typically be done at manufacturing time (in case of embedded devices), but in our case it done when the content master initializes.

Client Count	Registration Time (s)
1	1.2
10	38.6
50	240.3
100	515.1
200	1097.1

Table 2. Client registration times

The time needed for client registration is significantly longer than the previous case since there are several asymmetric cryptographic operations involved. The average registration time is around 4.1s.

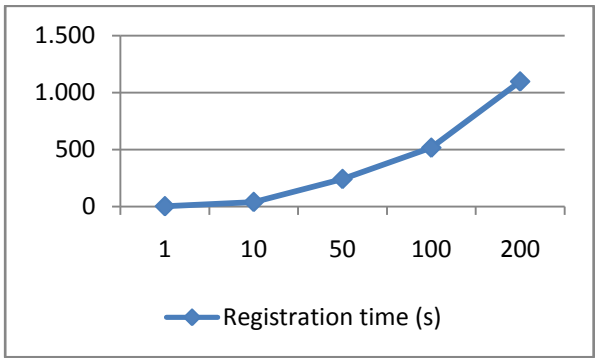

Fig. 5 Client registration time variation

Content registration is typically performed after content provider registration, because the content master will encrypt the content with a random key which will then be split among the registered providers.

Content Size (MB)	Registration Time (s)
0.5	67
1	98
5	580
30	2918
50	5525
100	8393
150	18775
200	24794

Table 3. Content registration times

The time needed for content registration grows linearly with size. The average registration time is around 110 ms per MB.

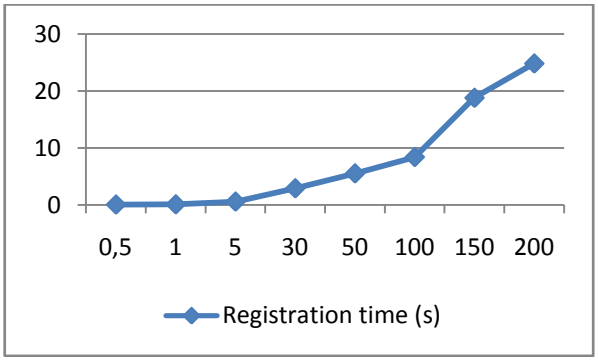

Fig. 6 Content registration time variation

B. *Content-related Operations*

Content distribution refers to sending the digital content to clients is an operation comprised of several tasks, such as:

- Content provider authorization request
- Session key generation
- Computation of the lock X
- Computation of the content hash
- Content encryption with the session key

We have run SCADDIC in a variety of conditions, ranging from 1 client with a mere 0.5MB piece of content to 25 clients and a 200MB piece of content. The times given below are for the broadcast preparation only (as described by the list of operations shown above) and do not contain the duration of the content transmission. The duration is dependent upon link speed and transient network conditions so we chose to benchmark only the parameters over which we can have control.

Clients/ Size(MB)	1	5	10	15	20	25
0.5	0.68	10.86	23.48	33.52	47.58	62.11
1	0.21	7.96	17.12	26.85	38.41	51.23
5	0.97	8.63	17.54	28.34	39.38	54.00
30	6.37	13.47	23.6	33.37	44.80	58.77
50	9.90	17.39	27.27	36.84	48.84	61.86
100	1.31	26.81	38.55	47.91	58.06	69.89
150	2.28	33.14	41.00	49.07	58.95	69.79
200	34.56	41.24	49.19	57.85	67.71	78.62

Table 4. Broadcast preparation time for various number of clients and content sizes

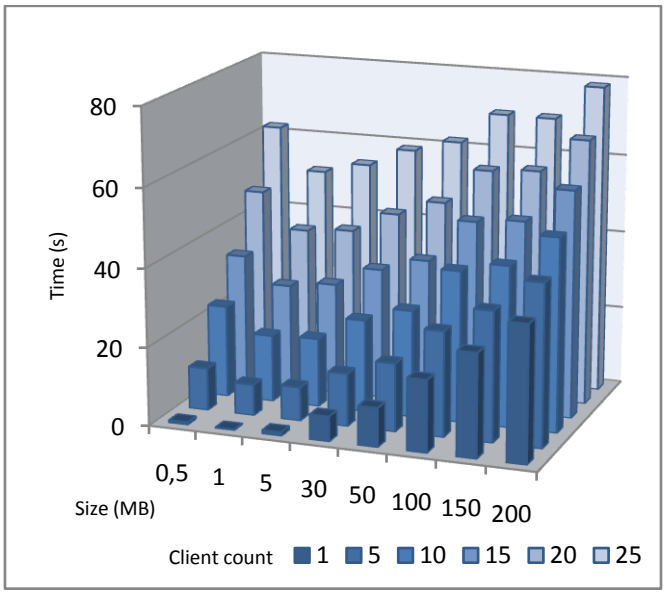

Fig. 7 Broadcast preparation time variation for various number of clients and content sizes

The analysis of the times needed to prepare the content for broadcast reveals that the time grows slowly with the number of clients and with minor influence from the size of the input content.

The average time to calculate the lock X for one client (without additional overhead) is around 117 ms.

Redistributing the content is an operation which essentially involves two parties: the sender and the receiver. In this case there is no need for the lock X to be computed, however, the content is encrypted with a session key which in turn is encrypted with the private key of the recipient.

Content Size (MB)	Redistribution Time (s)
0.5	0.05
1	0.09
5	0.45
30	2.57
50	4.53
100	8.98
150	12.69
200	16.04

Table 5. Content redistribution times

The time needed for content redistribution grows linearly with size. The average redistribution time is around 88 ms per MB.

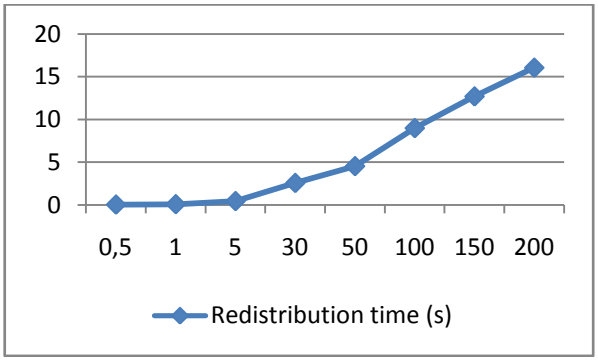

Fig. 8 Content registration time variation

V. Broadcast vs. Unicast Comparison

For the purpose of performance evaluation, the content distribution architecture can be summarized as the following actions:

1. Session key generation
2. Content encryption with the session key
3. Session key encryption (this is essentially creating the lock X which will later be used to recover the session key)
4. Distribution of the encrypted content and the encrypted session key

The content distribution time is then expressed by the following formula:

$$T_{dist} = \sum_{i=1}^{N} Tsk_i + \sum_{i=1}^{N} Tce_i + \sum_{i=1}^{M} Tx_i + \sum_{i=1}^{N} Tsend_i \quad (1)$$

Where:

- T_{dist} – Total time needed to transmit all instances of the content
- T_{sk} – Time needed to generate the session key
- T_{ce} – Time needed to encrypt the content with session key
- T_x – Time needed to generate lock X or to encrypt the session key with the public key of the recipient
- T_{send} – Time needed to transmit the content instance to the client
- N – number of instances of the content to be transmitted
- M – number of clients to which the content is to be transmitted

For the broadcast scenario, the number of instances of the content to be transmitted is N = 1, therefore equation 1 becomes:

$$T_{broadcast} = Tsk + Tce + \sum_{i=1}^{M} Tx_i + Tsend \quad (2)$$

Where:

- $T_{broadcast}$ – Total time needed to broadcast the content

For the unicast scenario, the number of instances of the content to be transmitted is equal to the number of clients, such that N = M, therefore equation 1 becomes:

$$T_{unicast} = \sum_{i=1}^{N} (Tsk_i + Tce_i + Tx_i + Tsend_i) \quad (3)$$

Where:

- $T_{unicast}$ – Total time needed to send the content in unicast mode

Looking at equations 2 and 3, it is clear that the broadcast scenario is far superior in terms of scalability. The question that remains to be answered is the order of magnitude for the difference. For this, we need to analyse the performance of the two scenarios varying the content size and the number of clients simultaneously requesting the same content.

For the purpose of this comparison we have made the following assumptions, derived from our benchmarking on the test machine:

- The session key is generated in 2 ms.
- Content is encrypted at a rate of 73 ms for 1MB, using a symmetric algorithm.
- The lock X takes 117 ms to compute for 1 client and exhibits a linear increase with the number of clients.
- For unicast, the lock X is not calculated and it is replaced by a single asymmetric encryption, which we consider to take 117 ms as well.
- The link speed between the sender and the receiver is 200 KB/sec.

The performance timings that we obtained as a result of several rounds of running the SCADDIC prototype are summarized below:

	0.5MB	1MB	5MB	30MB
Broadcast 1	3	5	26	156
Unicast 1	3	5	26	156
Broadcast 5	6	8	29	159
Unicast 5	14	27	131	780
Broadcast 10	14	17	38	168
Unicast 10	27	53	261	1559
Broadcast 50	295	298	318	448
Unicast 50	137	267	1305	7796
Broadcast 100	1173	1175	1196	1326
Unicast 100	273	533	2610	15593

Fig. 9 Performance for content sizes between 0.5Mb and 30MB

	50MB	100MB	150MB	200MB
Broadcast 1	260	519	779	1039
Unicast 1	260	519	779	1039
Broadcast 5	263	522	782	1042
Unicast 5	1299	2597	3895	5194
Broadcast 10	271	531	791	1050
Unicast 10	2598	5194	7791	10387
Broadcast 50	552	812	1071	1331
Unicast 50	12989	25972	38954	51937
Broadcast 100	1430	1689	1949	2209
Unicast 100	25979	51944	77909	103874

Fig. 10 Performance for content sizes between 50Mb and 200MB

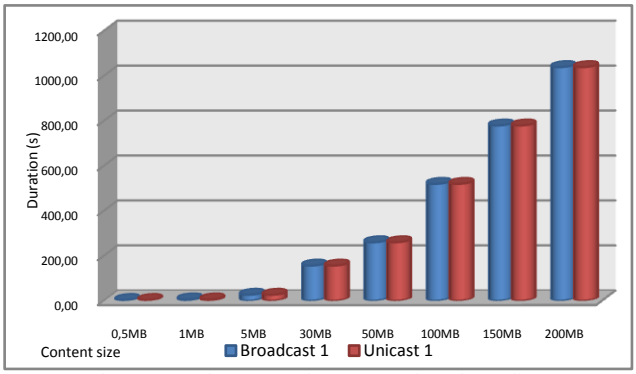

Fig. 11 Broadcast vs. Unicast comparison for 1 client

For a single client, the behaviour of the broadcast technique is the same as the unicast technique, so no speed improvement will be noticed. In fact, broadcasting the content purports having at least two clients requesting the same digital content therefore in case of a single client requesting a piece of digital content, broadcast has no benefit.

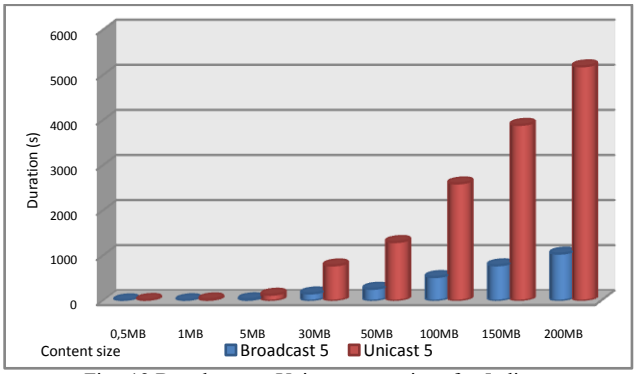

Fig. 12 Broadcast vs. Unicast comparison for 5 clients

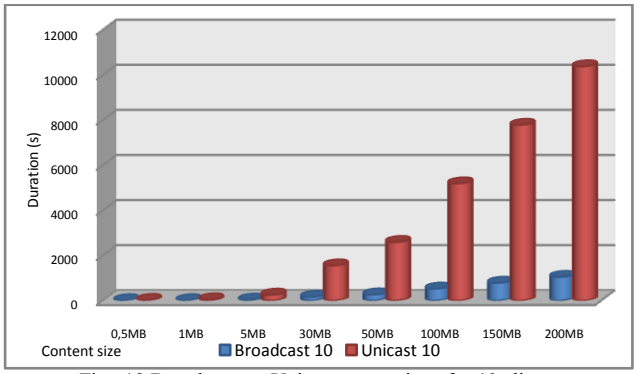

Fig. 13 Broadcast vs. Unicast comparison for 10 clients

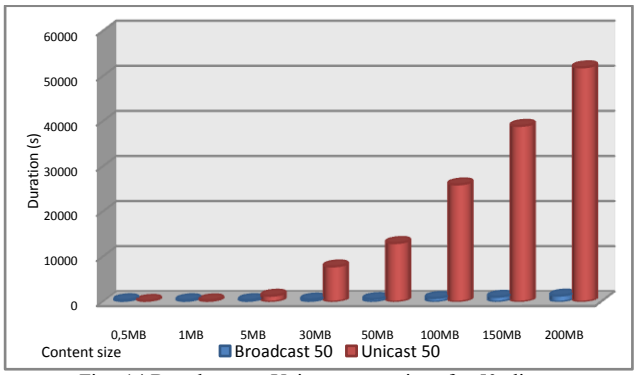

Fig. 14 Broadcast vs. Unicast comparison for 50 clients

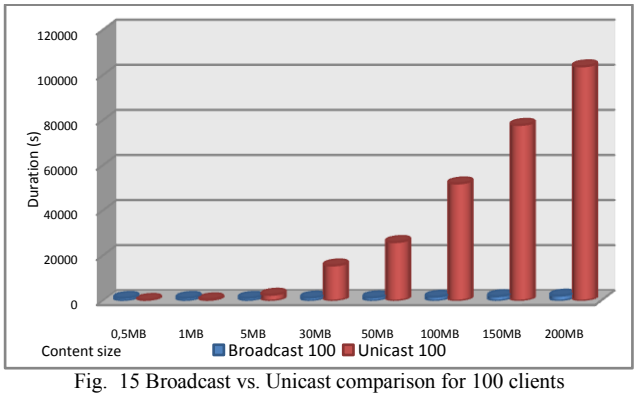

Fig. 15 Broadcast vs. Unicast comparison for 100 clients

Considering the link speed constant, the broadcast performance is directly proportional to the size of the content and inversely proportional to the number of clients serviced simultaneously.

VI. Conclusion

Digital content distribution by broadcasting and relaying content is a technique aimed at reducing the workload on the servers at moments of high demand. Depending on the content size, the number of simultaneous requests and speed link, broadcasting can be more scalable than the traditional

unicast approach. The three factors that we mentioned must be kept in balance, in order to get the most benefits out of content broadcasting.

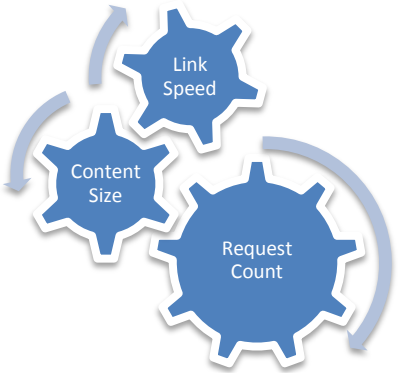

Fig. 16 Factors which affect the performance of the broadcast technique

Our implementation has shown that under circumstances, broadcast communication can have a tremendous speed and scalability advantages over the traditional unicast approach while preserving all the benefits stemming from rights management and content redistribution.

REFERENCES

[1] Valer Bocan, "Scalable and Secure Architecture for Digital Content Distribution", SoftCom International IEEE Conference on E-Commerce Technology, 2006

[2] Trusted Computing Group, "Trusted Computing Platform Alliance Main Specification", October 2003, Version 1.2, http://www.trustedcomputinggroup.org