New Channels, Old Concerns:
Scalable and Reliable Data Dissemination

Colin Allison, Duncan McPherson

School of Computer Science,
University of St Andrews,
KY16 9SS, Scotland

Dirk Husemann

IBM Research,
Zurich,
Switzerland

Abstract

An interesting trend in the continuing convergence of information technologies is the emergence of the Internet as a content provider in its own right, as opposed to its simply being one of many delivery channels. For example, it is increasingly the primary source for items such as court rulings and software releases. Unfortunately the IP protocols normally employed for reliable data transfer are of the point-to-point type and not well suited to large-scale one-to-many dissemination. Sudden rushes to obtain new items can cause severe traffic congestion and degrade network service across a whole region. Even worse, sites which are routinely popular cause routine congestion. Broadcast technologies should be able to provide a better solution in terms of scalability. The Internet has a mature protocol suite for IP multicast and more recently the traditional wireless broadcast industry has started moving from analog to digital transmission formats. However, in both these cases the emphasis in protocol development has been on support for continuous media, which requires timeliness of delivery rather than bit-perfect data integrity. A further problem with the new digital broadcast channels is their lack of support for integration with the Internet. This paper examines some of the issues involved in providing both reliable and scalable dissemination across broadcast channels and describes the DABWeb architecture for Internet content dissemination via digital broadcast.

1. Introduction

"Beyond the PC" invokes a vision of the near future where users are mobile and interact with attentive environments [1] directly through speech and gesture. Links with a variety of services will be setup through automatic exchanges between wireless information appliances using personal area networks such as Bluetooth [2]. These local, transient affinities will include links to wide area wireless multimedia services that offer content of interest to large numbers of users. What will that content consist of?

One trend is the emergence of the Internet as a source of content in its own right. Court rulings, government statements, fast breaking news that escapes local censorship, popular software releases, community information, and even accredited education programs are examples where the Internet is *the* primary source, as opposed to only being one of many delivery channels. Yet again, the medium has become the message. WebTV [3] was an early example of this convergence, raising the question of whether broadcast TV would become another application package on the desktop computer, or if the Internet would become another channel on the domestic television.[1] More recently, PCs are being used for global telephony [4] using the Internet as a carrier, while WAP enabled mobile phones offer access to Internet content.

The growing importance of Internet content is accompanied by a growing need for its scalable and reliable dissemination. Sudden rushes for newly posted items can cause very high traffic peaks. These surges in demand can rapidly cause server overload, even for major multi-server sites, and more seriously, cause widespread congestion that degrades service quality for unrelated IP traffic. The IP protocols typically employed for data dissemination – HTTP and TCP – are of the point-to-point type and poorly suited to cope with one-to-many scenarios when there are thousands of concurrent connections to a single source. The intrinsic reliability and politeness of TCP offers no protection against repeated broken connections caused by router overload and server failures. Web proxy caching and other types of replication can alleviate congestion caused by routine traffic to sites consisting exclusively of static

---

[1] The original WebTV product (1995) was a set-top box with no local storage, which connected to the Internet by a phone line. It involved Sony and Philips (end-system integrators); Excite (WWW search engine), Surfwatch (WWW filter); Concentric (ISP); IDT (electronics); Headspace(ambient music); and Progressive (audio/IP specialists). The company was bought by Microsoft at some point, and is now a brand name for a wider range of services and products.

HTML but cannot mitigate the problems caused by high peak request rates for dynamically generated content. At a fundamental level, the simplistic use of multiple point-to-point connections to meet a one-to-many requirement does not scale well. The following two sections examine two non-mutually exclusive alternatives (i) IP multicast and (ii) digital broadcast. Section 4 then describes the design and implementation of DABWeb, an architecture for Internet content dissemination via digital broadcast.

2. IP Multicast

IP multicast [5] scales better than any other IPv4 protocol, but scalability is a matter of degree and there are some longstanding drawbacks when looking to the Mbone for reliable data dissemination. It is a best-effort delivery protocol and the application protocols associated with it have mainly been concerned with continuous media [6], where timeliness of delivery is more important than data integrity. Many networking protocols use ARQ for reliability. The fundamental problem with using this class of protocol in a multicast setting is the "ACK implosion". A single source can send to tens of thousands of receivers without problem, but if they respond with ACKs or NACKs, the aggregate return traffic will cause severe congestion as it converges on the source. A variety of reliable multicast transport protocols have been developed [7], such as RMTP [8], which seek to reduce the implosion problem. These only scale to a limited extent however, reflecting the inevitable trade-off between reliability and scalability when feedback channels from receivers to sender are used.

A further consideration is that IP multicast is by no means universally available. Although it has been around for over ten years it is still not a required part of IP, and ISPs do not automatically support it, thus ruling out its use for a significant part of the Internet user population. IPv6 [9] makes multicast a standard part of IP, but IPv6 is not yet widely deployed.

Finally, the Mbone is embedded in the Internet, and can thus contribute towards congestion even in send-only mode unless special measures are taken. In [10] two techniques are suggested for making the Mbone "fair with TCP". Forward error correction using erasure codes [11], and controlling sender behaviour with TCP-like slow-start/ multiplicative-decrease. As the latter requires multiple feedback channels it is still a trade-off between reliability and scalability.

So, although IP multicast continues to have great potential, there is a growing recognition of the need for alternatives to the congestion-prone Internet for large-scale reliable data dissemination. Can advantage be taken of the scalability afforded by the new digital broadcast media?

3. Digital Broadcast

The broadcast industry is striving to change from analog to digital delivery mechanisms. The change is primarily motivated by the potential for better bandwidth utilisation (more channels) through the use of audio, image and motion compression. Further potential benefits include:

- once audio/visual content is digitally formatted it can be delivered through novel channels such as computer networks and portable digital media
- program associated data can be transmitted – e.g. no need to send a stamped, self-addressed envelope for a free accompanying booklet – a PDF file will be downloaded to your PC/TV/Radio/Phone appliance
- broadcast channels can be used for general-purpose data dissemination e.g. Usenet News, software updates, miscellaneous web content, etc.

The last point is of particular interest as it offers a potential solution to the scalability problems associated with distributing popular Internet content. There is no technical limit on the number of concurrent receivers of a wireless broadcast, within the area covered. The bandwidth required by the transmitter and the quality of reception are independent of the number of receivers. However, the communication asymmetry of digital broadcast raises other issues when it is used for data dissemination: *reliability*, *availability*, *service tuning* and *charging*.

3.1 Reliability

The encoding schemes used for continuous media, such as MPEG, are not suitable for bit-perfect data copying. ARQ protocols cannot be used in the absence of a back channel. This problem is further compounded in the case of mobile wireless receivers that are only intermittently connected. Two techniques can be employed in the absence of two-way communication to ensure reliable transfer of data: repeat transmission scheduling and forward error recovery.

The concept of a *data carousel* is used in broadcast networks to schedule repeated transmissions of the component parts of an object that should be received in its entirety e.g. a web site. Each slot in the carousel is loaded with a part of the object, including a CRC for error detection. The carousel is then revolved for some period of time, feeding the transmission network. Repeated transmissions increase the chance of a mobile user being able to complete a download despite interruptions, and for any client to recover from corrupt reception. The carousel is loaded to reflect scheduling policy. For example, in [12] the files used to represent a navigation bar are transmitted more often than other site components because of that item's importance.

Forward error correction (FEC) is not often employed in the Internet. The Internet protocols evolved against a patchwork of duplex networks where ARQ was a natural choice for achieving reliability. Furthermore, FEC codes are perceived as having significant bandwidth overheads and requiring complex algorithms that are costly to run in software. In broadcast networks, which are bandwidth rich and asymmetric, FEC becomes not only more attractive, but completely essential.

Erasure codes [11] are a type of FEC designed for one-to-many computer networking protocols, such as IP multicast. The basic idea is that k blocks of source data are encoded into $n > k$ blocks of transmitted data in such a way that any k blocks of encoded data can be used to recover the original data. So, a receiver can recover from up to $n - k$ losses of good blocks. A particular coding scheme is characterised as (n,k). For example, Reed-Solomon codes have effective limits of (64, 255) for efficient software encoding and decoding. The potential use of erasure codes as part of a congestion avoidance strategy for reliable bulk data transfer across the Mbone are described in [10], and (again in conjunction with IP multicast) form the basis of the Fcast application [13].

Interestingly, Rizzo's work on erasure codes [11] was motivated by a perceived gap between the telecommunications world, where FEC codes and hardware support are well developed, and computer networks, where there is a dearth of FECs suitable for use as software components in IP protocols. That convergence trend could usefully be extended to broadcast networks by combining erasure coding with carousel scheduling.

3.2 Availability

Unlike the Internet, a client cannot initiate a download at any time. The broadcast service provider determines in advance which information will be transmitted at which times. Selected sites can be repeatedly broadcast, reflecting updates and changes, using a data carousel. Client-side caching can store data and maintain the cache according to a policy that the user can influence, either explicitly or implicitly. Disk storage is currently plentiful and inexpensive, and newer storage technologies promise even more storage in less space.

3.3 Service Tuning and Charging

A conventional network server knows exactly how many requests are being made, for which files, at which times, and where they came from. How does a broadcast service collect this type of information? Conventional broadcast networks rely on surveys to assess the appropriateness of program scheduling. Similar techniques could be used to establish the success of data dissemination. Alternately, if a charging scheme is in operation then the revenue generated from each set of data broadcast is a measure of the demand. Subscription could be implemented by encrypting each data set, and using smart cards for decryption. The IBM Java Card [14] for example has a rich set of built-in cryptographic functions and can act as a tamperproof key for decrypting received data sets. The service provider primes the card with secret keys, one set for each service subscribed to, and sells it to the client.

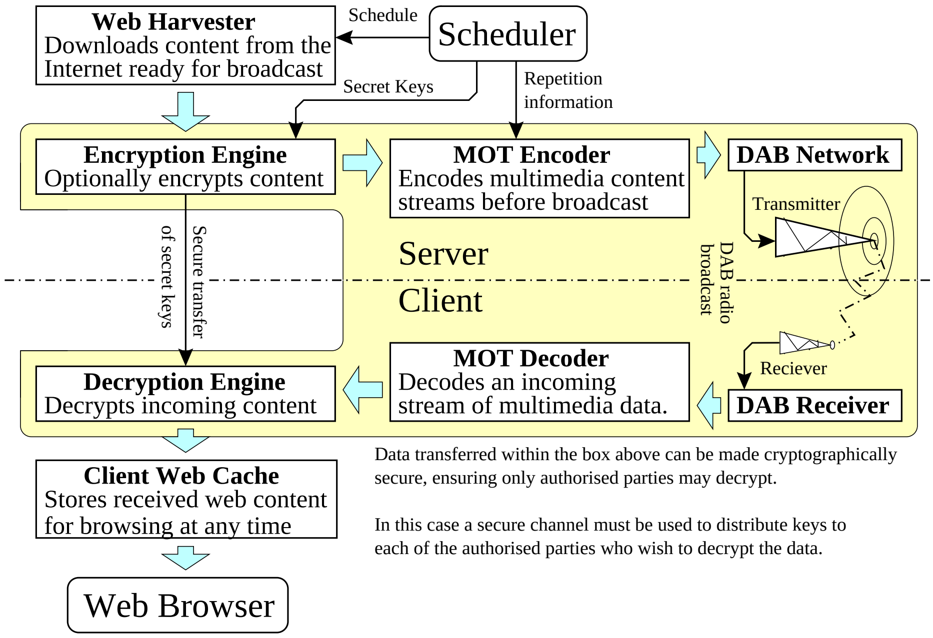

Figure 1: DABWeb System Overview

4. DABWeb: Digital Broadcast of Internet Content

DABWeb [15] is a model for the use of Digital Audio Broadcast (DAB) to disseminate Internet content. A reference implementation has been built and tested on a small DAB transmission network (with broadcasts limited to the laboratory), and a variety of receivers. DAB [16] is a set of standards for the transmission and reception of audio content and program associated data using digital wireless networks. Much of Europe is already covered by DAB. It is of interest in that it is a relatively stable set of standards that is being deployed in a variety of information appliances by a range of manufacturers. At present DAB receiver devices include car, domestic, and portable radios, and PCI and PCMCIA cards.

Figure 1 gives a system overview of DABWeb. The Web Harvester is based on WebFS [17], which represents web sites as conventional Unix file systems. The MOT (Multimedia Object Transfer) protocol was developed by ETSI [18] to standardise data transmission over DAB. MOT makes provision for data framing and repeats in a data stream to support carousel scheduling but does not include FEC coding. DABWeb clients enhance availability by downloading when possible, and maintaining a cache.

A DAB ensemble consists of up to 64 subchannels totaling no more than 1.87Mb/s. The ensemble is modulated by the transmitter using Orthogonal Frequency Division Multiplexing across a collection of frequencies totaling approx. 1.5MHz. Two modes of transport are possible for MOT data streams:

- Packet Mode where the MOT data stream occupies an entire DAB subchannel
- PAD (Program Associated Data) Mode where the MOT data stream shares a subchannel with audio data.

In PAD mode only 54 data bytes may be inserted per audio frame, at the rate of one frame per 24 milliseconds.

This provides a total bandwidth of approximately 2.19Kb/s on the DAB testbed in the laboratory. In Packet mode the bandwidth of a subchannel is dynamically configurable from the total 1.87Mb/s, by the multiplexer responsible for generating the ensemble. It is possible to dynamically create pop up subchannels after reducing the bandwidth allocation of existing subchannels. For example during news audio broadcasts where the audio quality required is lower, the bandwidth used by the news audio subchannel can be halved, and a new subchannel created to absorb the freed ensemble bandwidth. The new subchannel can carry an MOT data stream, and then disappear when the news finished.

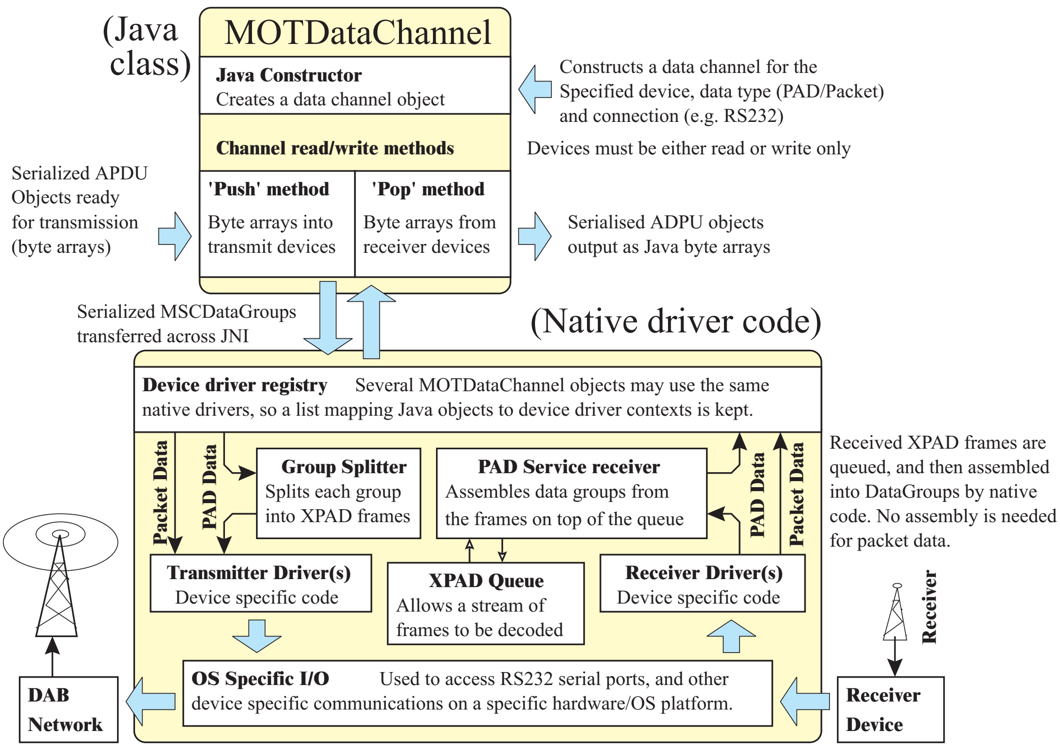

Figure 2: MOT Device Driver

4.1 Protocol Development in Java

Java was used for most of the implementation, allowing a modular object oriented approach. Java's own object serialization mechanisms were not used because (i) the resultant byte arrays created for Java objects did not fit the MOT standard, and (ii) the use of serialized Java objects at any level in the MOT protocol stack would require all clients and servers to use a JVM. Some coding in C proved necessary for device interfaces, and was integrated using the Java Native Interface. Figure 2 shows the heavy reliance on native code necessary for the MOTDataChannel driver.

4.2 Profiling and Subscription using Java Cards

Optional support for subscriptions and personal profiling is provided through the use of smart cards and cryptography. Customers are supplied with smart cards that decrypt selected parts of a broadcast. In contrast to the public key style of cryptography, a symmetrical Remote Key algorithm, BEAST RK [19], is used. Beast RK uses secret keys, a secure hashing function, and a stream cipher function. Each session key is randomly generated by the service provider. The DABWeb BEAST applet on the card decrypts the first 20 bytes of a stream to obtain the session key which is then passed to the host, (which typically has more processing power), where it is used as a stream cipher to decrypt the remainder of the session data.

At no point does the client know what the secret keys are. The time required to complete the decryption of an N byte stream is the sum of (i) the round trip time for the card to decrypt the first twenty bytes and return the result to the host, and (ii) the time taken by the host to decrypt the remaining N-20 bytes. This must be less than the time taken to transmit N bytes.

The IBM Java Card[2] used in the pilot is fully compliant with the OpenCard standard [20], which is also actively supported by other card manufacturers. Communication protocols between a host and a smart card are specified in ISO 7816. All data exchanged between the host and card (see Fig.3) takes the form of Application Protocol Data Units (APDUs). Each APDU is a byte array, with internal format specified for both commands and results.

The first communication between the host (running OpenCard) and a newly powered up JavaCard is the ATR which the card transmits. When the host receives the ATR byte array the lowest software layer of OpenCard is able to recognize the card type as being JavaCard. OpenCard then selects which JavaCard applet it will communicate with. This is done using a JavaCard OS APDU. Each applet on the card has a unique identifier number known as the Application Identifier (AID).

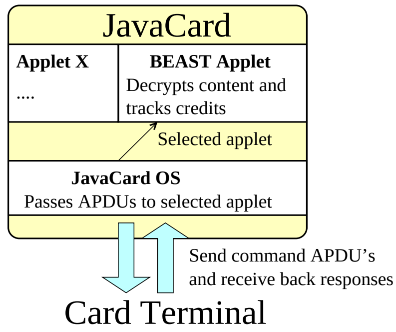

Figure 3: A Java Card with Two Applets

The AID is encoded in two parts, firstly a universally unique company identifier, assigned to the applet's developer, and also a number that uniquely identifies the applet, assigned by the developer. This allows any combination of applets that the card can accommodate to coexist.

A significant advantage of this type of multi applet card system is that a BEAST applet can coexist with a user's credit card applet, or along with other multimedia and e-commerce applets. With such a system, users potentially need only one smart card for all their needs.

4.3 DABWeb application areas

Any site hosting frequently requested Internet content could benefit from DABWeb. Sites that experience severe traffic congestion and server overload due to high peak demand would benefit most. Other envisaged applications of DABWeb involve support for remote and mobile users. To that end we envisage DABWeb devices communicating with close proximity devices using protocols such as Bluetooth. DABWeb appliances can provide a wireless application-level gateway between broadcast information services and other appliances in the near environment thereby contributing to the functionality of smart spaces. Electronic newspaper delivery, enhanced in-car information, community information, and distance learning programs are examples where a DABWeb extension to an existing Internet-based service would prove useful. In the case of an accredited degree course a smart card could be used to store learning credits and a students record. The card would only decrypt content appropriate to that student's course, and would securely maintain the results of a student's interaction with course materials. It is quite reasonable to consider the use of DABWeb for this type of purpose in a small country such as Scotland where DAB is under the control of a publicly funded broadcast company (the BBC) who have a remit to serve the community.

[2] At time of writing an IBM Java Card has 14KB EEPROM, 32KB RAM, 750 bytes Java heap, 160 bytes Java stack, garbage collection on EEPROM/RAM. Built-in cryptographic functions include DES, 3DES, SHA, RSA, DSA & RSA/DSA key generation.

5. Related Work

The Boston Community Information System [21] is an early example of using digital broadcast for reliable data dissemination. Two hundred PCs were equipped with packet radio cards, information was stored on several co-operating database servers and broadcast on a packet radio network. Users maintained personal profiles and interacted with the service by specifying queries that were sent to the servers via non-wireless network connections.

The Broadcast Disks project [22] has also investigated the use of data broadcasts in database applications. It has formulated a useful characterisation of broadcast services based on push/pull and periodic/aperiodic categorisation.

A variety of reliable multicast file transfer protocols have been developed, including the Multicast Dissemination Protocol [23] and the Adaptive File Distribution Protocol [24]. These both combine IP multicast with some form of moderated ARQ. Fcast [13] distinguishes itself by using erasure codes rather than ARQ, which makes it potentially more scalable in error-prone environments.

6. Conclusion

Several new phenomena have been identified in the emerging digital landscape – the Internet as a content provider in its own right, increasingly sophisticated smart cards, and the availability of novel digital audio broadcast channels. These trends are complimentary in a variety of ways. The Internet does not cope well with peak demands for its own content whereas any number of receivers in a given region can tune in to a wireless broadcast without impacting on bandwidth requirements or delivery quality. Smart cards can be used to match clients needs with broadcast scheduling. A design and reference implementation for supplying Internet content via digital broadcast – DABWeb - has been briefly described. DABWeb has successfully integrated several component technologies, including OpenCard, the BEAST cryptographic system, WebFS, Java, and DAB. DABWeb appliances may also communicate with nearby appliances using low powered wireless systems such as Bluetooth. In this case DABWeb can provide a channel between wide area information sources and the local environment.

At present MOT carousels in DABWeb provide no forward error correction and rely entirely on repeat transmission scheduling for reliable delivery. The issue of whether erasure codes can significantly enhance the performance of the MOT protocol warrants further investigation.

7. References

1. Spohrer, J. and M. and Stein, *User Experience in the Pervasive Computing Age*. IEEE Multimedia, 2000. **5**(1): p. 12-17.
2. Bluetooth, *http://www.bluetooth.com*. 1999.
3. Microsoft, *WebTV: http://www.webtv.com*, 2000.
4. Polyzois, A.C., *et al.*, *From POTS to PANS: A Commentary on the Evolution to Internet Telephony*. IEEE Network, 1999. **13**(3): p. 58-63.
5. Deering, S., *Host extensions for IP multicasting*, 1989.
6. McCanne, S., *Scalable Multimedia Communication-Using IP Multicast and Lightweight Sessions*. IEEE Internet Computing, 1999. **3**(2): p. 33-45.
7. Obrczka, K., *Multicast Transport Protocols: A Survey and Taxonomy*. IEEE Comms. 1998. **36**(1): p. 94-102.
8. Lin, J.-C. and S. Paul. *A Reliable Multicast Transport Protocol*. in *IEEE Infocom*. 1996: IEEE Press.
9. Bradner, S. and A. Mankin, *The Recommendation for the IP Next Generation Protocol*, . 1995: RFC-1752.
10. Vicisiano, L. and J. Crowcroft. *One to Many Bulk-Data Transfer in the Mbone*. in *HIPPARCH'97*. 1997. Uppsala, Sweden.
11. Rizzo, L., *Effective Erasure Codes for Reliable Computer Communication Protocols*. ACM Computer Communication Review, 1997. **27**(2): p. 24-36.
12. Fuhrhop, C., A. Kraft, and R. Kubis, *Object Carousel Simulator for Broadcast Applications*, in *Multimedia'99*, N. Correia, T. Chambel, and G. Davenport, Editors. 1999, Springer-Verlag: New York. p. 83-92.
13. Gemmell, J., J. Gray, and E. Schooler, *Fcast Multicast File Distribution*. IEEE Network, 2000. **14**(1): p. 58-69.
14. Baentsch, M., *et al.*, *JavaCard – From Hype to Reality*. IEEE Concurrency, 1999. **7**(4): p. 36 - 43.
15. McPherson, D.F., *Support for Internet Information Services on Digital Audio Broadcast Networks. Master's Thesis*. School of Computer Science, University of St Andrews. 2000,
16. Eureka_147_Partners, *Radio Broadcasting Systems: Digital Audio Broadcast to Mobile, Portable and Fixed Receivers*, 1997, European Telecommunications Standards Institute: France.
17. Zwahlen, P., *Design and implementation of a secure web-based file system, Masters Thesis*, Eurecom Institute. 1999.
18. ETSI, *DAB Multimedia Object Transfer protocol.*, 1998.
19. Lucks, S. *Beast: A fast block cipher for arbitrary block sizes*. in *IFIP Conference on Communications and Multimedia Security*. 1996: Chapman & Hall.
20. Husemann, D. and R. Hermann, *OpenCard: Talking to Your Smartcard*. IEEE Concurrency, 1999. **7**(3): p.53-57.
21. Gifford, D., J. Lucassen, and S. Berlin, *An Architecture for Large Scale Information Systems*. ACM OSR (Proc. of the 10th SOSP), 1985. **19**(5): p. 161-170.
22. Acharya, S., *Broadcast Disks: Dissemination-based Data Management for Asymmetric Communication Environments*. Ph. D Thesis, Brown University, 1998.
23. Macker, J. and W. Dang, *The Multicast Dissemination Protocol Framework*. IETF Draft, 1996.
24. Cooperstock, J.R. and S. Kotsopoulos. *Why Use a Fishing Line When You Have a Net? An Adaptive Multicast Data Distribution Protocol*. in *USENIX'96*. 1996.