DYNAMIC IMAGES CAROUSEL TRANSMISSIONS OVER MULTIPLE MULTICAST GROUPS

SUPAKORN RATSAMEEMONTHOL

UNIVERSITI SAINS MALAYSIA

2004

DYNAMIC IMAGES CAROUSEL TRANSMISSIONS
OVER MULTIPLE MULTICAST GROUPS

by

SUPAKORN RATSAMEEMONTHOL

Thesis submitted in fulfillment of the requirements
for the degree of
Doctor of Philosophy

December 2004

Dedicated to my parents for their patience and devotion, and to my wife for her encouragement.

ACKNOWLEDGEMENTS

I would like to convey my sincere thanks and deepest gratitude to my supervisor Assoc ***Prof Dr. Sureswaran Ramadass*** who introduced me to the field of Multimedia Networking. I am grateful to him for his encouragement and all the help and guidance provided to me, and for having taken his valuable time during the preparation of this thesis. His dedication will never be forgotten.

I am grateful to, School of Computer Sciences, Institute of Postgraduate Studies (IPS), the University Library and Universiti Sains Malaysia (USM) for providing assistance. I would like to convey special thanks to the staff of the Network Research Group (NRG), especially Leena Shankar for her help, support and an enjoyable time during my study.

I owe thanks to many people in Thaksin University who have been supportive and helpful, especially, Associate Prof. Pirod Intarasirisawat, the President of the university and Mr. Nipat Intong for supporting the scholarship.

Further, I would like to express my thanks to my fellow graduate students in the Postgraduate Lab., my dearest friends in Malaysia for their immense loyalty, love, fun, whole hearted support and encouragement during this adventurous period of my postgraduate studies. I would not have enjoyed it without you ─ Adib, Mosleh, Aiman, Isabelle, Rusdi and Ibrahim

I finally want to thank Kulrapat, my lovely wife, and my parents, my brothers and my sisters for their encouragement. They were extremely patient and understanding during all these years. I would not have done my work without their support.

Supakorn
December 2004

TABLE OF CONTENTS

	Page
ACKNOWLEDGEMENTS	iii
TABLE OF CONTENTS	iv
LIST OF TABLES	vii
LIST OF FIGURES	viii
LIST OF ABBREVIATIONS	xiii
ABSTRAK	xv
ABSTRACT	xvii
CHAPTER 1 - INTRODUCTION	
1.1 Approaches to Dynamic Images transmission	2
1.1.1 Image Bit Plane Partitioning approach	3
1.1.2 Images Streaming Partitioning approach	4
1.1.3 Scan Line Partitioning approach	5
1.1.4 Tile Partitioning approach	5
1.2 Problem definition and proposed solution	6
1.3 Contributions of the thesis	10
1.4 Outline of the thesis	11
CHAPTER 2 - LITERATURE SURVEY	
2.1 Dynamic Images Transmission Protocol (DITP)	13
2.2 Data Carousel Scheduling	18
2.2.1 Method of scheduling data	19
2.2.2 The proposed updating technique	21
2.3 Packet Scheduling	24
2.3.1 Destination Set Splitting (DSS)	25
2.3.2 Multiple Channel Multicast (MCM)	25
2.3.3 Multiple Multicast Groups (MMG)	28
2.3.4 Generalized Multiple Multicast Group (GMMG)	29
2.4 Conclusion	32

CHAPTER 3 - DYNAMIC IMAGE TRANSMISSION PROTOCOL ENHANCEMENT

3.1 Image compression format enhancement 34
- 3.1.1 Compressed image size 34
- 3.1.2 Compressed image legibility 36

3.2 The proposed optimal tile size 38
- 3.2.1 The examination of image overhead and compressed tile size 39
- 3.2.2 Selecting tile size 41

3.3 The proposed temporal redundancy removal 43
- 3.3.1 Rectangular temporal redundancy removal 43
- 3.3.2 Tile comparison problems and solutions 45

3.4 Conclusion 49

CHAPTER 4 - DATA CAROUSEL SCHEDULING

4.1 The proposed Data Carousel Scheduling 52
- 4.1.1 Scheduling components 53
- 4.1.2 Data Carousel Scheduler 55
  - 4.1.2.1 Incorporating of tiled images 56
  - 4.1.2.2 Scheduling outgoing tiled images 58
- 4.1.3 The Immediate Cycle Restart with Object Reordering and Re-updating (ICR/RR) updating proposed technique 62
  - 4.1.3.1 ICR/RR Algorithm 62
  - 4.1.3.2 Retransmission control for the old updated tiles 65

4.2 Conclusion 67

CHAPTER 5 - OBJECT-ORIENTED MULTIPLE MULTICAST GROUPS (OOMMG) TRANSMISSION SERVICE

5.1 The proposed OOMMG transmission service 70
- 5.1.1 Definition of the signalization information 72
- 5.1.2 State machine of the OOMMG scheduler 72
- 5.1.3 The general processing of the OOMMG scheduler 73

5.2 The OOMMG updating proposed technique 77
- 5.2.1 Rate assignment with cumulative scheme 78
  - 5.2.1.1 Equal cumulative scheme 78
  - 5.2.1.2 Doubling cumulative scheme 79
  - 5.2.1.3 Multiplicative cumulative scheme 80
- 5.2.2 The OOMMG updating algorithm 81
  - 5.2.2.1 Base level updating 82

5.2.2.2 Level N updating 82
5.3 Transmission rate control 89
5.3.1 Flow-in controlling 90
5.3.2 Flow-out controlling 91
5.4 Conclusion 93

CHAPTER 6 - EXPERIMENTAL RESULTS AND DISCUSSION
6.1 Experimental methodology of the new DITP enhancement proposed technique experiment 96
6.2 Experimental results of the new DITP enhancement proposed technique 97
6.3 Experimental methodology of the OOMMG 99
6.3.1 The experimental methodology of $Q_{\rightarrow in}$ 101
6.3.2 The experimental methodology of $Q_{waiting}$ 102
6.4 Experimental results of the OOMMG 102
6.4.1 The experimental results of Q_{in} 103
6.4.2 The experimental results of $Q_{waiting}$ 104
6.5 Discussion of the results 105
6.6 Conclusion 107

CHAPTER 7 - CONCLUSIONS AND FUTURE DIRECTIONS
7.1 Conclusions 109
7.2 Future Directions 113

BIBLIOGRAPHY 115

APPENDICES

APPENDIX A - EXPERIMENT RESULT OF CHAPTER 3 120
APPENDIX B - SIMULATION RESULT OF CHAPTER 4 128
APPENDIX C - EXPERIMENT RESULT OF CHAPTER 6 129
APPENDIX D - STATE MACHINE NOTATION 134
APPENDIX E - SIGNALIZATION MESSAGE 135
APPENDIX F - VIRTUAL TIME CONVERSION 136

LIST OF PUBLICATIONS AND SEMINARS 137

LIST OF TABLES

		Page
Table 4.1	Description of state of Data Carousel Scheduler	56
Table 4.2	Description of event of Data Carousel Scheduler.	56
Table 5.1	Description of state of the OOMMG scheduler.	73
Table 5.2	Description of event/action of the OOMMG scheduler.	73
Table 5.3	Description of state of the source pointer traversing SM.	85
Table 5.4	Description of event/action of the source pointer traversing SM.	85
Table 5.5	Description of decision condition of the source pointer traversing SM.	86
Table 5.6	Description of transition of the source pointer traversing SM.	86
Table 5.7	Description of state of the destination pointer traversing SM.	86
Table 5.8	Description of event/action of the destination pointer traversing SM.	86
Table 5.9	Description of decision condition of the destination pointer traversing SM.	87

LIST OF FIGURES

		Page
Figure 1.1	A Progressive Image Transmission Capability System (from Lalich-Petrich et al. 1995).	4
Figure 2.1	The original real-world image (a). and the image after being tiled (b).	15
Figure 2.2	A Flat Broadcast Program (adapted from Acharys et al., 1995a; 1995b;1996).	19
Figure 2.3	A Multi-disk Broadcast Program (adapted from Acharya et al., 1995a; 1995b;1996).	20
Figure 2.4	Carousel updating Techniques (from Buchholz et al. 2001).	22
Figure 2.5	A Destination Set Splitting multicast schedule (from Ammar and Wu 1992).	25
Figure 2.6	An example of Session Organization scheduling (from Donahoo et al. 1999).	27
Figure 2.7	An example of Partition Organization Scheduling (from Donahoo et al. 1999).	27
Figure 2.8	An example of an MMG scheduling approach with three layers (from Bhattacharyya et al. 1998).	29
Figure 2.9	GMMG scheduling in case there are some remaining packets (from Roca 1999b).	30
Figure 2.10	GMMG scheduling of an isolated ADU (from Roca 1999b).	31

Figure 3.1 JPEG/JPEG-2000 comparisons for real-world images with PSNR = 24.6: a) 15.5KB JPEG image (10% quality factor) and b) 13.4KB JPEG-2000 (1.5% quality factor). 35

Figure 3.2 JPEG/JPEG-2000 comparisons for text images with PSNR = 19.9: a) 29.7KB JPEG image (10% quality factor) and b) 18.4KB JPEG-2000 (2% quality factor). 36

Figure 3.3 Compression efficiency results of JPEG and JPEG-2000 in real-word image. 37

Figure 3.4 Quality factor results of JPEG and JPEG-2000 in real-world image. 37

Figure 3.5 Quality factor results of JPEG and JPEG-2000 in text image. 38

Figure 3.6 Compression efficiency results of JPEG and JPEG-2000 in text image. 38

Figure 3.7 JPEG and JPEG-2000 overhead comparison for small tiles. 39

Figure 3.8 The comparison between JPEG and JPEG-2000 compression rate for real-world image. 41

Figure 3.9 The comparison between JPEG and JPEG-2000 compression rate for text image. 41

Figure 3.10 The structure of matrix R that contains of the status of 256 tiled images. 44

Figure 3.11 The mapping views of applications A, B, C, D and E to matrix R and the capture region of the views. 44

Figure 3.12 Comparison of memory usage of CRC method and P-P method. 46

Figure 3.13 Pixel position shows in partly of a 48x64-pixels tile. 48

Figure 3.14(a) Comparison time of CRC method and P-P method before improving with 50% pixels comparison method, 240x320-pixels image used. 48

Figure 3.14(b) Comparison time of CRC method and P-P method after improving with 50% pixels comparison method, 240x320-pixels image used. 49

Figure 4.1 The Data Carousel Scheduling. 52

Figure 4.2 State Machine of Data Carousel Scheduler. 55

Figure 4.3 The scheduling of sending carousel program P_t in virtual time scale unit. 57

Figure 4.4 The examples for explaining the interaction between the scheduler and the outgoing buffer. 61

Figure 4.5 The examples of the carousel program from the updating process. 61

Figure 4.6 The updating region of continuous and intermittent updating. 62

Figure 4.7 The examples of intermittent updating. 64

Figure 4.8 The examples of continuous updating. 65

Figure 4.9 The simulation of comparison between equation 4.4 and 4.5 67

Figure 5.1 The OOMMG transmission service system. 71

Figure 5.2 State machine of the OOMMG scheduler. 73

Figure 5.3 The example of the OOMMG scheduling queue. 74

Figure 5.4 The example of an equal cumulative scheme transmission. 78

Figure 5.5	The example of doubling cumulative scheme transmission.	79
Figure 5.6	The example of a multiplicative cumulative scheme transmission.	80
Figure 5.7	The initiating of the Flowing-Control-Object with doubling cumulative scheme and the submitting of 16 ADUs.	81
Figure 5.8	Flat updating algorithm of the base level.	82
Figure 5.9	The components of the level N updating algorithm.	84
Figure 5.10	State machine of the OOMMG updating technique	84
Figure 5.11	The updating new ADU-Data to level 2.	88
Figure 5.12	The result of updating the submitted ADU-Data to the scheduling queue.	89
Figure 5.13	Spreading the transmissions to reduce burstiness. (a) Spreading at each tick time. (b) Spreading at each of two tick times.	91
Figure 5.14	The example of Pre-fragmentation method.	93
Figure 6.1(a)	Capture processing time.	97
Figure 6.1(b)	Comparison processing time.	98
Figure 6.1(c)	Compression processing time.	98
Figure 6.2	Number of tile of the partitioning 240x320-pixels image.	99
Figure 6.3	Compression time of 240x320-pixels real-world image and text image	99
Figure 6.4	Duplication ratios as a function of the Q_{in} size and transmission level, 256 ADUs burst.	103

Figure 6.5 End of reception times as a function of the Q_{in} size and transmission level, 256 ADUs burst. 104

Figure 6.6 Duplication ratios as a function of the $Q_{waiting}$ size and transmission level, $Q_{in} = 1$. 104

Figure 6.7 End of reception times as a function of the $Q_{waiting}$ size and transmission level, $Q_{in} = 1$. 105

LIST OF ABBREVIATIONS

ADU	: Application Data Unit
ADUs	: Application Data Units
ALF	: Application Level Framing
CC/ER	: Cycle Completion with Early Removal
CRC	: Cyclic Redundancy Check
DSS	: Destination Set Splitting
FCC	: Full Cycle Completion
FEC	: Forward Error Correction
FIFO	: First In First Out
Gbps	: Gigabits per second
GMMG	: Generalized Multiple Multicast Groups
HTML	: HyperText Markup Language
HTTP	: Hypertext Transfer Protocol
ICR	: Immediate Cycle Restart
ICR/R	: Immediate Cycle Restart with Object Reordering
ICR/RR	: Immediate Cycle Restart with Object Reordering and Re-updating
IP	: Internet Protocol
IPv6	: Internet Protocol version 6
JPEG	: Joint Photographic Experts Group
JPEG2000	: Joint Photographic Experts Group 2000
LDCF	: Largest-Delay-Cost-First
LZW	: Lempel-Ziv-Wellsh
MBone	: Multicast Backbone
Mbps	: Megabits per second
MCM	: Multiple-Channel Multicast
MDP	: Markov Decision Process
MMG	: Multiple Multicast Groups
MTU	: Maximum Transmission Unit
NIC	: Network Interface Cards
OOMMG	: Object-Oriented Multiple Multicast Group
OS	: Operating system
PDU	: Protocol data unit
PDUs	: Protocol data units
PO	: Partition Organization channel scheduling

PROTRAC	: Progressive Image Transmission Capability
PSNR	: Peak Signal-to-Noise Ratio
QoS	: Quality of Service
SM	: State machine
SO	: Session Organization channel scheduling
T1	: T-1 carrier
TCP	: Transmission Control Protocol
UML	: Unified Modeling Language
VIC	: Video Conferencing Tool
WWW or W3	: World Wide Web

PENGHANTARAN COROUSEL IMEG DINAMIK MELALUI BERBILANG SALURAN BERKUMPULAN

ABSTRAK

Pertumbuhan Internet yang medadak serta kemajuan yang pesat dalam teknologi komputer-meja tahap tinggi telah memudahkan penyebaran data dan multimedia pada skala yang lebih meluas. Seiring dengan itu, permintaan kepada servis seperti ini dan isipadu jalur lebar talah meningkat secara berterusan. Keperluan aplikasi multimedia terhadap keupayaan menghantar data masa-nyata sejajar dengan pertumbuhan skala dan kepelbagaian, telah meransang penyelidikan dalam struktur penghantaran data, penskedulan, penghantaran data dan servis transmisi skala besar. Khidmat ini melibatkan transmisi 'carousel' imej yang dinamik kepada kumpulan multi-tebar berganda. Servis transmisi ini menawarkan tiga tuntutan seperti yang telah diterangkan sebelumnya. Pertama, ia menyediakan struktur imej dinamik yang berskala. Imej yang besar tidak akan dihantar secara serentak, sebaliknya ia akan disusun dalam petak-petak kecil bersaiz jubin sebelum disebarkan kepada ramai pengguna. Imej dalam saiz yang sesuai ini akan dianalisa dalam menentukan imbangan antara masa pemprosesan pembuangan sementara dan bebanan trafik yang disebabkan oleh imej sementara. Pengekod imej juga akan diuji. Tuntutan kedua ialah, ia menyediakan daya tahan kepada tranmisi data berulang. Teknik 'carousel' data mempunyai kelebihan iaitu ia dapat membasmi urusan memgendalikan pertanyaan daripada para pelanggan yang terlalu ramai tentang kehilangan paket. Proses memgemaskini data, menyusun petak-petak imej ke atas kitaran 'corousel' adalah kunci utama dalam teknik ini. Melalui skim kemaskini yang kami cadangkan, ia mengambil masa kependaman yang pendek untuk menghantar data berbanding dengan teknik lain. Selain dari itu, produk sampingan dari kaedah ini juga mampu menangani permintaan penerima yang lewat.

Tuntutan ketiga, servis transmisi menyediakan keskalaan dan kepelbagaian. Rangkaian multi-tebar menyokong protokol keskalaan ini. Keperbagaian penerima telah dicapai melalui penghantaran kepada kumpulan multi-tebar berganda. Kunci utama daripada servis ini ialah menskedul petak-petak imej kepada kumpulan multi-tebar dan mengendali susunan data yang dihantar. Melalui pengunaan teknik berorientasikan objek, kaedah ini dapat membantu memudahkan transmisi petak-petak imej yang agak luarbiasa. Tambahan iagi, kadar kawalan trafik menjadi lebih lancer dengan mengawal data bertindih yang disebabkan penskedul algoritma dan data multi-tebar berskala yang menghasilkan data pra-serpih. Akhir sekali, usaha kami menyumbang ke arah penyebaran imej dinamik berskala besar kepada pelbagai ciri penerima yang besar jumlahnya, dengan ketahanan yang disediakan oleh transmisi berulang.

DYNAMIC IMAGES CAROUSEL TRANSMISSIONS
OVER MULTIPLE MULTICAST GROUPS

ABSTRACT

The skyrocketing growth of internet and the advancement of high-end desktop computers have facilitated multimedia data distribution in large-scale. Congruously, the demand of this kind services and the bandwidth volume grow endlessly. The requirements of multimedia applications for the capability in transmitting real-time data with scalability and heterogeneity have motivated the research into data delivering structure, data delivering scheduling and large-scale transmission services. This work proposes a dynamic images carousel transmission over multiple multicast groups. The transmission service offers three demands that were described previously. The *First*, it provides a scalable dynamic images structure. Instead of sending a big image all at once, the image is tiled into small parts before being distributed to a large number of receivers. The suitable tile size will be analyzed in order to balance between the temporal removing processing time and the traffic load caused by the temporal image. The image encoders also will be tested. The *second* demand is that it prepares a cyclic data transmission for resilience. This data carousel technique gains the benefit of eliminating the management of the explosion request for lost packets from the huge number of clients. The process of updating data, tiled images, into the carousel cycle is the main key of this technique. With this proposed updating scheme, it takes a short latency time to retransmit the data compared to other techniques. Moreover, the by-product of this method serves the late arrival of receivers. The *third* demand, the transmission service provides scalability and heterogeneity. The multicast network supports the scalable protocol. The heterogeneity of receivers has been fulfilled by transmitting data over multiple multicast groups. The key point of this service is the method to schedule tiled image data over multicast groups and to handle the sequence of sending data. Using object-oriented technique to implement such method helps to

ease the transmission of the peculiar tiled images. Moreover, the traffic rate control has been achieved by controlling the duplication of data caused by a scheduling algorithm and periodical multicasting data that conveys a bulk of pre-fragmented data. Finally, our work contributes to the large-scale distributing dynamic images to the large number of dissimilar kinds of receivers with resiliency that provide by cyclic transmission.

CHAPTER 1

INTRODUCTION

Over the years, the process in computer networks and computer technologies has resulted in the rapid growth of multimedia applications. In 1969 the Internet worldwide has grown-up from 4 hosts to more than 56 million today (Zakon 2004). Computer technologies have experienced important developments that can nowadays enable real-time multimedia processing in personal computers. These technologies have recently crossed a threshold that has made dissemination of multimedia applications feasible at a reasonable quality.

Multimedia applications require a high communication bandwidth and substantial data processing more than any other internet application. For example, a quality video conference requires a high bandwidth and compression to be able to be transmitted on today's typical user connections. There are three conflicting requirements: bandwidth, processing power, and quality. All multimedia applications require minimum bandwidth consumption, processing level, and an acceptable level of media quality. Conversely, a lower bandwidth can only be achieved at the expense of a higher compression processing, which leads to a longer delay. If we want to keep the compression process low, the reliability is decreased because of packet loss from the network congestion. A good tradeoff between the two main technology factors, processing and bandwidth, has enabled multimedia applications with reasonable quality.

There has been much research and development of protocols and services for video and audio transmissions, less work has been focused on the transmission of shared data generated by the applications in real time. While sharing the view of an application can be thought of as a kind of video streaming or mutable dynamic images, the unique characteristics of these images require new algorithms and transmission protocols to achieve legibility and dynamic size.

The main objective of this thesis is to provide solutions to the major problems in the dissemination of the dynamic images that are generated on-the-fly. The main problems are:

i- The processing time of the image tilling and compression.

ii- The quality of the images.

iii- The consumption of the network bandwidth and the reliability of the transmission.

The proposed solutions are:

i- To enhance the tiling scheme of an image and the compression of the tiled images to optimize the processing time required.

ii- To study the image compression formats that provide the best quality of the compressed images.

iii- To propose an approach for the transmission of data using the Data Carousel scheduling method over multicast channels. This method will be able to control the transmission rate of the data and to retransmit the data in order to ensure reliability.

1.1 Approaches to Dynamic Images transmission

As stated above, the running applications create a lot of views which can be thought of as a kind of video streaming, or in other words, it can be thought of as a series of still images that are sent continuously. Transferring an image takes orders of magnitude more bandwidth than any other data type. However, image data is suitable for compression which will results in the reduction of its size while maintaining the quality. Multimedia applications and other tools that use image data are usually not scalable enough with even improved protocols (Lane 1994; Ruiu 1994) and other hardware enhancements. The proposed thesis is focused on resolving these issues and emphasizes on the approach of partitioning images into smaller items in order to send them more efficiently and to allow scalability.

There are several ideas of image partitioning techniques: (a) Image bit plane partitioning approach, (b) Image streaming partitioning approach, (c) Scan line partitioning approach and (d) Image tiling approach. The following section will introduce briefly on these approaches.

1.1.1 Image Bit Plane Partitioning approach

This approach (Lalich-Petrich et al. 1995) (PROTRAC) allows progressive and scalable images to transfer over the World Wide Web (WWW or W3). At the server-side, the original image will be partitioned along its bit plants. The 8 bit gray scale image will amount to 8 planes. In the case of a true color (24 bit) image, the same procedure is applied for each of the three bands (RGB) producing 24 slices. Naturally, each plane will consist of zeros (0) and ones (1) only. The Lempel-Ziv-Wellsh (LZW) (Nelson 1989) image encoder has been applied to the planes after 8 consecutive bits of one plane into a byte. Alternatively, the original image can be averaged (i.e. over regions of size SxS) before applying the slicing procedure. This will reduce the amount of data to be transferred, but it will cause a poor image transfer. At the client-side, when a bulk of data arrives, it is added to the existing image data (if available). If the arriving data are in the compressed image bit planes format, then they will be decompressed and pasted on “Top” of the planes currently defining the image. The additional bit plane provides a better color or gray scale resolution as resolution is increased by a factor of two. The client controls the maintenance of the order which the bit planes are composed in order to represent an image. Figure 1.1 depicts the structure of the system.

From the communication point of view, the system transfers images over the World Wide Web, using the TCP protocol. No image data is sent twice through the network. Therefore, the bit plants will be sent in order over reliable transmission.

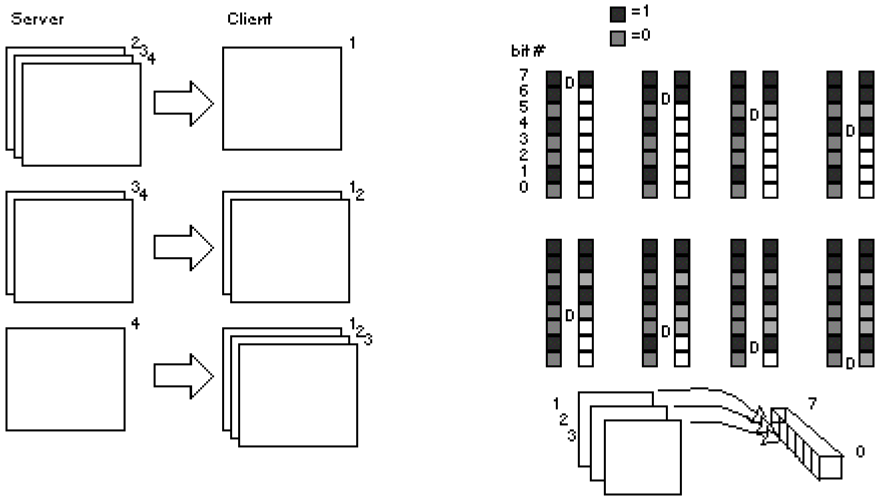


Figure 1.1: A Progressive Image Transmission Capability System (from Lalich-Petrich et al. 1995).

1.1.2 Images Streaming Partitioning approach

Deshpande & Zeng (2001) proposed a scalable architecture for streaming the JPEG-2000 images, using the Hypertext Transfer Protocol (HTTP). The approach used JPEG-2000 to compress the original image into multiple independent portions. The byte-range feature of the HTTP streaming concept (Fielding et al. 1999) has been used for streaming the data portion. By using the byte-range feature, the web server can host large images and can also stream to the client. Consequently, the client can view an image at a variety of resolutions and quality levels. It is also possible to stream only a selected region of the image at a particular resolution and later decompress this partial stream and display it. Thus client devices with different capabilities, variety of screen resolutions and heterogeneous bandwidths can all achieve a scalable viewing of the same content stored in a single file.

1.1.3 Scan Line Partitioning approach

This approach (Richardson et al. 1998) is proposed to solve the requirement of the thin client that requires more communication and interaction with the server. They are simpler than thick clients and can be downloaded quickly from the internet. This method moves the execution of the applications completely to the server, leaving the client stateless. When the applications are executed at one site, their frame buffers or the screen images are partitioned along its scan line and are transferred across the network to be displayed at a remote-side. The examples of using this approach are Web-based Learning (Li et al. 1999) and Interactive Remote Instruction (Al-Shaer et al. 1997). Both of these have been used in the virtual classroom. Furthermore, the Scan Line Partitioning approach has been used for commercial applications such as WebEx Communications and Anabas Collaboration.

1.1.4 Tile Partitioning approach

The main objective of this approach (González et al. 2001) is to distribute the shared views being generated by a running application on the desktop computer. The views are supposed to be a stream of still images. With this method, at first, the views which are displayed on the desktop screen are captured as a still image. It is then partitioned into square tiles and each tile will be compressed to reduce its size by using the standard image compression technique. The dimension of the tile and the image compression ratio are the important factors to determine whether a tile can be transmitted by one network packet so that it will not cause packet fragmentation (Tanenbaum 1981). After the compressed tiles are packed into the multicast network packets, the packets are sent and repeatedly broadcasted over the IP multicasting. The advantage of the repeated broadcast is that the system will be able to overcome any transmission losses and provide the leaving and joining of participants seamlessly.

1.2 Problem definition and proposed solution

As described in the previous section each method has different ideas in the transmission of large images, but all have the same strategy, that is multiple division and conquest. Even though all of these approaches seem promising, they still have some inner problems. The transmission of images using these approaches can be considered in two cases. First, we will consider the transmission of two different types of images: the single still image and the dynamic image by using the IP multicasting[1] (Marcus & Kitty 1998) rather than the TCP's[2] reliable transport protocol (Postel 1981). Second, we consider the resiliency and scalability of multicast transmission.

The first consideration

The first type of image (still image)

- ***Using the image bit plane partitioning approach,*** a single still image is partitioned into multiple planes. They will be packed into the IP multicast packets and sent out. Since the IP multicasting does not send out the packets in sequence, all received packets of each plane must be accumulated first before displaying the image progressively. The system, accordingly, must buffer the fore-coming packets that lead to memory usage and management problems.
- ***Using the images streaming partitioning approach,*** each compressed image portion which is sent by using HTTP byte-ranges feature will be

[1] Multicasting is inherently unidirectional and does not require that a connection between two nodes to be established before the data transfer can begin. As the multicasting eliminates the time TCP takes to set up a connection, so it is better suited for time critical applications. Sending the multicast packet is analogous to broadcasting radio signals on a particular frequency. Just as you must tune a radio receiver to the particular frequency to receive the radio signal, you must join a multicast group to receive the multicast packets.

[2] The two main weak points of TCP protocol are the high response times and small scalability. TCP sets up a connection between two hosts before data transmission begins. The protocol, thus, takes a longer transmission time. Also, the connection models a reliable, bi-directional circuit, so as to guarantee that the data arrive with error-free and in order; hence, TCP is small-scale connection.

packed into several IP multicast packets. But it still has a memory problem like the first approach.

- ***Using the scan line partitioning approach,*** a single still image is sent in a left-to-right scan line order over the IP multicast packets, but at the receiver-side they arrived un-orderly, causing the same problem as the previous approaches.
- ***Using the tile partitioning approach,*** the memory problem is overcome by assigning one IP multicast packet to one compressed tile image. As a result, the receiving packets can be displayed immediately without buffering although they are not received orderly.

The transmission of a single still image as expressed previously indicates that in the first three approaches, even though the image is partitioned into independent small parts, they will become dependent on each other again when they are delivered via IP multicasting (one part per several packets) as compared to the last approach (one part per packet).

The second type of image (dynamic image)

The transmission of a dynamic image is more complicated than the transmission of a single still image. The first three approaches take a high processing cost in determining the temporal redundancy[3]. For example, the system must index the altered pixels so that the receiver-side can determine how to display them. As compared to the last approach, only the altered tiles will be transmitted without indexing of the altered pixels.

[3] Several parts of the image that remain unchanged from one immediate of time to the next called temporal redundancy. The detected parts that remain unchanged are not retransmitted.

The proposed solution in this thesis applies the image tiling approach to construct a new dynamic image transmission system. We use a new standard image compression, JPEG-2000 (ISO/IEC FDIS15444-1:2000), which supports not only the real-world images but also the text images. To minimize the processing time of removing the temporal redundancy, the exploration of a suitable tile size and the method to optimize the processing time of comparing the image need to be considered. It will be described in more detail in chapter 3.

The second consideration

Resiliency problem

Resiliency refers to the ability to overcome transmission losses and to accommodate the leaving and joining of receivers. A Data Carousel is applied to solve the resiliency at the application level to transmit the tiles in a cyclic fashion. Using the carousel scheduling with the dynamic images requires a method that would be able to update the tiled dynamic images in carousel cycle.

Scalability problem

Scalability provides the ability to reduce the connection response time by eliminating the need for acknowledgements or any feedback from the receivers. Applying the IP multicasting to improve the scalability will lead to the problem of how to manage the heterogeneous receivers. Several methods have been proposed to address the heterogeneity. Our work relies on the packet scheduling scheme. For example, Destination Set Splitting (DSS) (Ammar & Wu 1992); (Ammar & Towsley 1997), Multiple Multicast Groups (MMG) (Bhattacharyya et al. 1998), Multiple-Channel Multicast (MCM) (Donahoo et al. 1999; Vicisano 1998) and Generalized Multiple Multicast Groups (GMMG) (Roca 1999c). The packet scheduling schemes have been designed to handle interdependent data[4]. This will lead to the problem of scheduling the tiled images because each tiled image is the independent data.

The proposed solutions to overcome resiliency and scalability are.

Resiliency solution

The new Data Carousel enhancement proposed in this thesis provides an updating method that incorporates the new tiled images into the carousel cycle. This updating method contributes to two advantages. The first advantage provides a fair transmission between the new tiled images and the old updated tiles. The second advantage provides a cyclic updating to ensure more resiliency.

Scalability solution

The new GMMG enhancement proposed in this thesis uses the Object-Oriented method to manage the multicast channels into the objects of the multiple multicast channels. The tiled images of a carousel cycle will be incorporated into these objects and sent to the receivers. The advantage of this proposed solution is that the tiled images can be transmitted in a sequence from the top-left corner of the image to the right-bottom corner. For example, the text-based images need to be sent in a sequence so that it will be displayed at the receiver-side in a way that the user can quickly read the content of the image.

[4] Interdependence means that the content of the sending packets is related to each other.

1.3 Contributions of the thesis

This thesis contributes the following.

- We provided the technique of rectangular tiling scheme. This scheme shows the optimal tile size that provides two advantages. First, it will minimize the number of compression of tiled images. Second, it will minimize the number of comparison of the tiles when removing the temporal redundancy of the image.
- We proposed an approach for the Data Carousel scheduling method based on a Flat Indexed Data Carousel, and we demonstrated the capability of updating the tiled images into the carousel cycle. With this updating approach, we demonstrated how it is able to solve the retransmission of the lost data thus providing resiliency.
- We proposed a multicast object scheduling for high transmission rates to solve the high-end heterogeneity receivers. This method of scheduling has shown its ability to transmit the tiled images (independent data) in a chronological order, and has shown the rate control to regulate the stream of tiled images from the Data Carousel scheduling to the multicast object scheduling in order to control the duplication of the sending data. Also, we have presented the rate control to adjust the flow of the sending data. Within the experiments conducted, two main parameters (the end of reception times and the duplication of the sending data) are tested and their results are analyzed. The analyzing results provide important information to optimize the efficiency of the multicast object scheduling.

In this thesis, there is a formal study on the related operations and the computations of their complexity. The idea is to focus on measuring both accuracy and efficiency. The implemented system is based on the image tiling scheme, carousel scheduling and multicast object scheduling.

1.4 Outline of the thesis

This chapter provides an introduction to the work by giving a brief overview of the state of the art and the related concepts, including a brief summary on the work conducted here.

In chapter 2, we study other related concepts used in this thesis. The image partitioning method, Data Carousel scheduling and packet scheduling will be discussed and the general problems highlighted.

In chapter 3, we introduce the concept of dynamic image transmission and present the relationship between dynamic images and video. Some examples of applications that can use dynamic images will be introduced. The focus of chapter 3 is on exploring the image compression format and the optimal tile size and analyzing the tiled images in both spatial redundancy removal and temporal redundancy removal.

Chapter 4 is devoted to describe an approach for the Data Carousel scheduling based on a flat indexed data carousel for the purpose of reliable transmission. The main point of this chapter is the novel updating technique. Furthermore, we present a method to regulate the flow of outgoing data that interacts with the network transmission service.

Chapter 5 presents an approach for the multiple multicast group transmission. The beginning of this chapter explains the infrastructure of this approach. A description of the update technique of the proposed method and various components, associated with examples to illustrate the process of updating data over the multiple multicast channels will constitute the main part of this chapter. Moreover, we describe the scheme to control the traffic rate of both the updating rate and transmission rate.

Chapter 6 is dedicated to two experiments. The first experiment will investigate the processing time of the new dynamic image enhancement of chapter 3. The second experiment will test the data transmission of chapter 5. These experiments have been done at the sender-side to acquire accurate results. In the second experiment, the main factors will be investigated and the results have been analyzed to find out a way to enhance the performance of transmission.

To conclude, Chapter 7 will give a brief summary of the main idea discussed in this thesis. Suggestions on future work are also found in this chapter.

CHAPTER 2

LITERATURE SURVEY

In this chapter, we will describe the state of the art of some major concepts used in this thesis. First, we shall describe the "Dynamic Images Transmission Protocol" (DITP) which is the tile partitioning scheme proposed. Second we shall introduce the carousel scheduling technique that will support the retransmission of the tiled images. Third, we shall explain how the packet scheduling technique will broadcast the sending tiled images over the multicast network to the heterogeneous receivers. In the discussion, we shall focus on the problems in the DITP and in both scheduling techniques and shall highlight the solutions of our proposed approach.

2.1 Dynamic Images Transmission Protocol (DITP)

Many specialized multimedia collaborative systems such as co-browsers (Bisdikian et al. 1998; Davis et al. 1997) and sharing tool engines (Abdel-Wahab & Feit 1991; Richardson et al. 1998) have been developed for distributing shared data over the network on-the-fly rather than sending hard copies or faxing the materials to remote participants. The major features of these systems in a collaborative session is to be able to distribute data information in real-time and to emulate a virtual projection screen or documents on a virtual table. With these features, it has made the original electronic form of a document not only a faithful version but also a compact representation. In summary, all the systems described above have the same desire in sharing a common view that can be overcome by transforming a visual view to an image and streaming related images if the view changes dynamically.

One can ponder on the question: why can't the image stream be sent with the existing video protocol and tools? Actually, there are some research done in using such protocol and tools. The first example, Lawence Rowe, from the University of California in Berkeley, has been using video technology to transmit data information via MBone (Eriksson 1994). They used a scan converter to transform the computer screen signal into a standard video format and a digital camera to capture the hard-copy slides. Then, the converted images were encoded with a H.261 (ITU-R Recommendation 1993) format before transmitting them out. The second example, the Network Research Group at University College, London has developed a video conferencing tool called the VIC version 2.8 to send data contents through video streams. The sender can select a region on the screen to capture frames that are different from the video frames. These selected portions, which are the visual views of the sender, will be shared by participants via transmission of such portions to all of them.

Even though the video approach as mentioned in the previous paragraph has achieved the requirement of distributing data, still this technique has some drawbacks:

- Predefined video dimensions depend on the video compression, so the video dimensions are limited to the dimensions of information to be shared whereas the use of converters through the entire display view will cause the view to be a public view.
- The required bandwidth for video transmission. The changes of video frames need more bandwidth that cannot be well managed by video compression techniques.

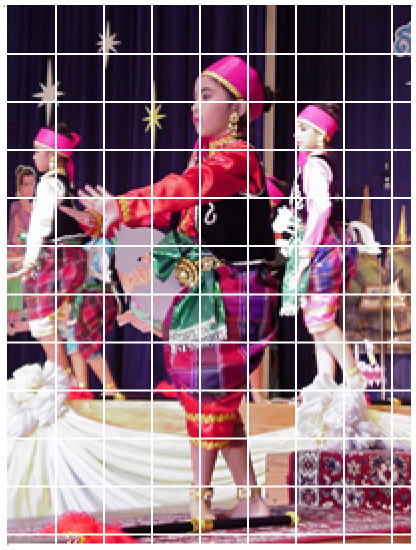
To solve the drawbacks, González (2001) has proposed an approach for transmitting dynamic images which is called the DITP. This protocol is used in collaborative systems using a visual view of the information to be shared with the participants. We can assume such images as the highest level representation of any information materials to be used for the discussion.

This protocol can be explained in two parts: image processing and image transmission.

Image processing

- ***Spatial redundancy:*** The DITP has been used for transferring streams of still images continuously. The running applications that create a lot of views can be viewed as a video stream of dynamic images. The spatial redundancy is the inevitable specifics of the dynamic images. Breaking the image into small blocks, for example, the video techniques of H.261 and H.263 (Côté 1998) is the algorithm of the still image compression to remove the spatial redundancy. In the DITP, each of the still image frame of dynamic images will be tiled into square blocks as demonstrated in Figure 2.1 and each block will be compressed by using a standard image compression to remove the spatial redundancy.

(a)



(b)

Figure 2.1: The original real-world image (a) and the image after being tiled (b).

- ***Temporal redundancy:*** Motions will occur when the image containing the dynamic graphics behave like video or movable regions. It can be assumed that the pixels within the current picture can be thought of as a translation of those of the previous picture. The DITP chooses motion prediction with a null motion vector because of the high cost of the computation, thus it gets an advantage from the tiles that remain unchanged from one sample to another. Consequently, the detected tiles that remain unchanged from one sample to another are not transmitted. This condition is only feasible during the processing of the dynamic images. Another advantage is that these images can be of a higher resolution than the traditional video images.

Image transmission

By using an unreliable transport protocol in the DITP, when the network is congested, the router scheduling will fragment the large packets. Thus, whenever a single fragment is missing, the entire large packet will be lost. The largest internet packet which is guaranteed not to be fragmented is 576 bytes (536 bytes actual data + 40 bytes standard header). The appropriate tile size and the image compression format are the factors of this limitation. The DITP selects 40(height) x 40(width)-pixels[1], the biggest tile that does not lead to packet fragmentation on the Ethernet.

In scheduling each altered tile over the multicast network, the tile needs to be sent once; however, it will be retransmitted at random times. After a random time, the protocol will send a refresh for each Packet Data Unit (PDU) in order to accommodate the late comers and to strengthen the protocol resiliency.

There are some issues that should be considered in this protocol which may limit their applications in the development of our approached system, such as:

i- ***Image quality:*** The DITP protocol uses two kinds of images[2] (the real-world image and the text image) with different sizes (40x40-pixels to 1024x678-pixels). Both of these images have different characteristics and different effects of

[1,2] with 24 bits per pixel and 16 million colors

compression. The sharp edges created by the text and geometric shapes would appear fuzzy and murky. It will get worse if the compression rate is higher. An example: the JPEG compression system is “Lossy”, meaning that whenever an image is compressed and decompressed, it loses its data. This eventually affects the image quality and can result in the distortion of the picture.

ii- ***Optimal tile size:*** The compression size and times, and network fragmentation are the important factors to define the optimal tile size. The first factor, if the small tile size is chosen, the comparison times and compression times will be longer compared to the bigger tile size. On the other hand, for the bigger tile size, the temporal redundancy will increase because it is hard to remove the redundancy. Due to the second factor, a tiled image must be compressed into one limited packet size, so the quality of the outcome image will be decreased.

iii- ***Temporal redundancy:*** The removal of the temporal redundancy method of the DITP is quite simple. The unchanged tiles can only be determined by comparing every pixel with its corresponding one in the previous sample, so this operation takes a lot of time and memory. These times and the sampling time will determine the maximum sampling rate. This protocol solves this problem by using the statistical comparison and sub-sampling comparison to speed up the comparison task. However, the overhead of the comparison’s calculation and the reliability of comparison must be considered.

The solutions for the limitations of a transmission of the dynamic images using the DITP are:

i- Study to find out the suitable image compression format that gives the best quality in compression of the real-world and text images.

ii- Study the compression of various sizes of the tiled image and find out the optimal tile size that gives the most effective temporal redundancy removal and the tile’s processing times.

iii- Provide a new method to solve the comparison of the tiled image problem including *i)* memory consumption *ii)* comparison time and *iii)* reliability of comparison by enhancement from the comparison method of the DITP.

2.2 Data Carousel Scheduling

To provide reliability for the transmission data, a common method is to use the ARQ, automatic request for retransmission. With the ARQ, the receivers use a back channel to the sender to send requests for retransmission of lost packets. The ARQ works well for one-to-one reliable protocols and has also been an effective reliability tool for one-to-many reliability protocols and in particular for some reliable IP multicast protocols. However, for one-to-very-many reliability protocols, the ARQ has limitations, including the feedback implosion problem because many receivers transmit ARQ back to the sender. Due to this there is a need for a back channel to send these requests from the receiver. Another limitation is that the receivers may experience different loss patterns of packets. Thus the receivers may be delayed by the retransmission of packets that other receivers have lost but they have already received. This may also cause a wasteful use of the bandwidth used to retransmit packets that have already been received by many of the receivers.

The Data Carousel (Acharya 1995b) method is proposed to overcome the above problems. With this method, the sender partitions the object into equal length pieces of data, places them into packets, and then continually cycles through and sends these packets. The receivers continually receive packets until they have received a copy of each packet. The Data Carousel has the advantage that it requires no back channel because there is no data that flows from the receivers to the sender and also provides "Late arrival" mechanisms. For example, all the receivers can start asynchronously to receive all the data. There are two parts that need to be considered the Data Carousel technique for distributing the dynamic images: the method of scheduling data and the updating technique.

2.2.1 Method of scheduling data

There are two methods to schedule data in a carousel cycle: the flat method and the non-flat method.

Flat Method

In this method, each data cannot be broadcasted more than once in one cycle. Figure 2.2 illustrated a flat data carousel style. In the figure data item A, B and C are broadcasted cyclically. The following are some applications:

i- The Boston Community Information System, described by Gifford (1990), is an early effort in data broadcasting. It applies an FM channel to broadcast the news and other information to radio receivers equipped with a personal computer. However, the data set must know in advance, unlike the dynamic image that is generated in real-time.

ii- Datacycle (Bowen et al. 1992) architecture is designed to exploit VLSI data filters to allow the client to process the database queries on a cyclically broadcast database. A drawback of this technique is the unlimited amount of data of the internet for broadcasting an entire database; consequently this may cause an inefficient use of bandwidth.

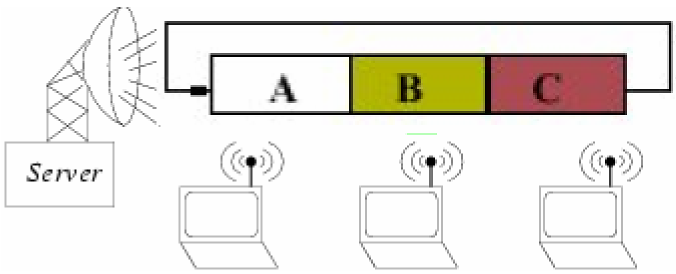


Figure 2.2: A Flat Broadcast Program (adapted from Acharya et al. 1995a; 1995b;1996).

Non-Flat Method

In this method, the data in the cycle can be broadcasted more than once. The Broadcast Disks (Acharya et al. 1995a; 1995b;1996) is an example of this method. The Broadcast Disks applies the Data Carousel to disseminate the database information. The key feature of the Broadcast Disks is the application that is called the Multilevel Disks. The data items that are most likely to be of interest to a larger part of the client community are broadcasted more frequently than others. The data broadcast with the same frequency is considered as belonging to the same disk. Thus, it can be said that Multiple Disks of different sizes and speeds can be laid on the broadcast medium. All clients can possibly access the data; the server determines the optimal percentage of the broadcast bandwidth to be allocated to each item. Then, the broadcast program is generated randomly in such a way that the average interval time between the two instances of the same item matches the requirement of the clients. Due to variance in the interval times, the expected delay of an item cannot be minimized. Figure 2.3 shows a simple example of a Broadcast Disks approach. In the figure, data item A is broadcast twice as often as B and C because the disk that contains A is more popular than the disk of B and C.

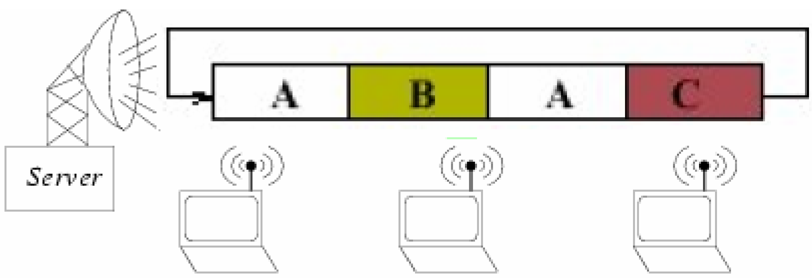


Figure 2.3: A Multi-disk Broadcast Program (adapted from Acharya et al. 1995a; 1995b;1996).

The new Data Carousel Scheduling enhancement proposed in this thesis use a Flat Data Carousel method. This method will send the tiles in sequence. For example, when a group of tiles of text image is broadcasted, the tile in the upper left corner is sent first, then the tile next to it on the right until the end with the tile in the bottom right corner. With this sending pattern, the text images will be displayed at the receiver-side in a way that enables the user to easily read the context in such an image. Therefore, we do not use the non-flat method because we cannot categorize each tile into groups.

The advantage of using the Flat Data Carousel is that it provides retransmission of the lost data. If a receiver loses a packet in one round of transmission, it can receive that packet again in the next round. But this simple retransmission cannot be used with the dynamic images that are sent in real-time. The following proposed updating technique overcomes this problem.

2.2.2 The proposed updating technique

Generally we consider the altered tiled images which are created by the DITP and are submitted by the application as a collection of messages, called the object set. Updating in the object set that need to be broadcasted occurs whenever the system determines to replace one or more of the objects. We use the term “Carousel update technique” in order to explain how to incorporate those updates into the carousel cycle. However, we assume that every object has been broadcasted at least once before the system will update the carousel cycle. The objective of an updating technique scheme is to reduce transmission delays not only for the newly updated objects but also for object that already been broadcasted. Some of the proposed updating techniques of the Flat Data Carousel are illustrated in Figure 2.4 (Buchholz et al. 2001). The details of each technique with their problems are described as follows:

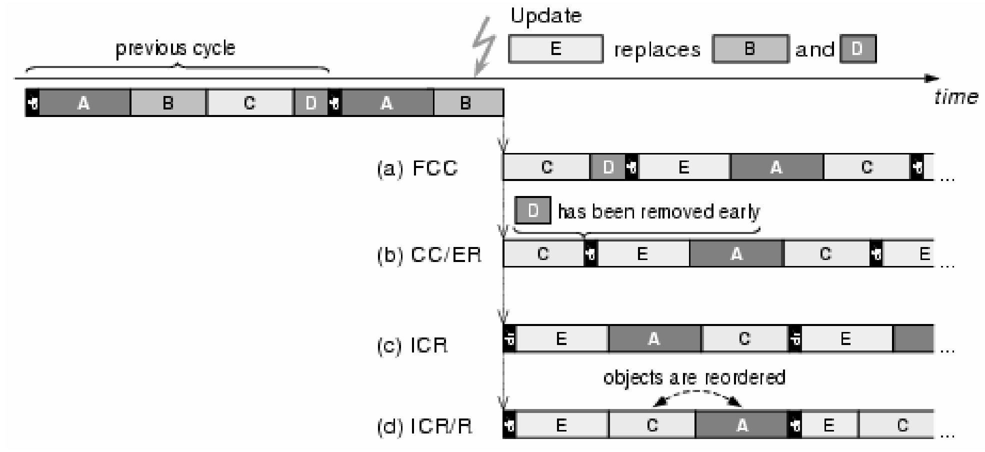


Figure 2.4: Carousel updating techniques (from Buchholz et al. 2001).

- **Full Cycle Completion (FCC)** as shown in Figure. 2.4 (a), the running cycle is fully completed before the beginning of a new cycle with an updated object set. To minimize the response delay time, new objects are inserted at the beginning of the next cycle. The disadvantage of FCC is the long latency time. It results from updating the object set and incorporating the update into the carousel cycle (depending on the cycle length).
- **Cycle Completion with Early Removal (CC/ER)** as shown in Figure 2.4 (b), this technique overcomes the disadvantage of FCC. However, CC/ER is related to FCC, all replaced objects are pulled out from the current cycles except those that have not been sent before. For this scheme, the completion time of the current cycle will minimize the latency between the update and the incorporation. On the other hand, some objects which have been declared by the directory are not broadcasted. This is the disadvantage of this approach.
- **Immediate Cycle Restart (ICR)** as shown in Figure 2.4 (c), this technique decreases the latency between the update and the incorporation. With the ICR, a new cycle begins after the instant termination of the current carousel cycle. As the transmission of a single object is considered to be an atomic operation, the termination must occur after the transmission of the currently sent object is

accomplished. The new cycle has the updated carousel directory at the beginning, together with the newly added objects and the objects of the previous cycle in an arranged order. A major problem in the ICR is that frequent updates will also cause frequent terminations, and hence those objects at the end of the cycle will possibly not be broadcasted at all.

- **Immediate Cycle Restart with Object Reordering (ICR/R)** as shown in Figure 2.4 (d), this technique solves the ICR problem. The objects that have never been broadcasted before are inserted at the beginning of the cycle and sorted in a chronological order according to their arrival. They are accompanied by the objects whose last transmission had the longest time and these objects are reordered based on their last broadcast time. The objects that were broadcasted immediately before the restart are at the end of the cycle. However, if the number of bytes of the newly updated objects are more than or equal to the number of bytes of the currently sending objects, the ICR/R will revert to the ICR's problem. Moreover, this problem is significant if we consider that the response time of the retransmission of the lost objects is important.

The new updating technique proposed in this thesis will overcome the problems of the retransmission time and the ICR/R described as follows.

Proposed retransmission time solution

Our updating approach provides the new retransmission method by dividing the tiled images of the carousel cycle into 3 types. The first type consists of the newly altered tiled images that will be updated at the beginning of the cycle. The second type also comprises of the tiled images derived from the first type of tiled images that have been broadcasted. Such tiles will be updated immediately again at the beginning of the cycle to provide the retransmission of the lost tiles. The third type comprises of the old tiled images that have not been broadcasted for a long time or the tiled images that have been broadcasted more than two rounds. The third type of tiled images can be transmitted again on behalf of the updating technique.

Proposed ICR/R solution

The proposed updating technique overcomes the ICR/R problem by incorporating some of the old tiled images to a group of tiled images currently being updated into the carousel cycle.

2.3 Packet Scheduling

Most of the internet applications comprises of clients requesting the existing data from a server. If the number of clients is big, the server encounters scalability. Multicasting the data to the set of clients (Wong & Ammar 1985; Clark & Ammar 1997; Acharya et al. 1995b; Almeroth et al. 1998; Mahajan et al. 1997; Nonnenmacher et al. 1997) is the solution for the server to handle a large volume of requests. A problem that arises when the data is broadcast is the heterogeneity of receivers. Heterogeneity, variation of network bandwidth and receiver capability causes the sender into deciding how to setup the transmission rate. If the sender uses a high sending rate to support the high-end receivers, the low-end will lose some data because of the limitation in receiving the data. In contrast, the high-end will receive data at the same rate as the low-end when a slow sending rate is applied.

Several ways have been proposed to transmit the data to a set of heterogeneous receivers within several multicasts network groups. Some of them rely on packet scheduling schemes (DSS/MCM/MMG/GMMG), others on coding (e.g. the hierarchical coding of a video stream (Bolot & Turleti 1994)), on transcoding (the information is tailored, e.g. an image definition is reduced according to the needs of a particular client (Fox et al. 1998)), or on filtering (Luby et al. 1999), (Wittmann & Zitterbart 1998) but we are particularly interested in the packet scheduling schemes in