Simulation and Comparison of Cost, Priority and FCFS Scheduling Schemes over an Adaptive Network Model

Bala P Ramaraju
Department of Computer Science
University of Illinois at Chicago
Chicago, IL 60607

Mitchell D.Theys
Department of Computer Science
University of Illinois at Chicago
Chicago, IL 60607

Abstract

Scheduling Data Requests is a highly researched topic in the field of Networks. Many optimal Scheduling Algorithms have been developed and effectively implemented. The advent of the Internet has brought revolutionary changes in the design of the Schedulers. The manner in which data gets carried over the networks has also changed. No longer is the Internet thought of as a Network used only by Intellectuals. It has become the common mans gateway to the world. With the advancement of technology the Internet became the carrier of a plethora of different Data types that include Video, Audio, Voice, Graphics, etc.

To support these high-end applications significant changes have been made to the way traffic is handled over the Internet. Scheduling has become very crucial to use the Network Bandwidth effectively. Some applications require a specific amount of Bandwidth to be reserved for them at scheduled times. Prioritization of Client Requests has to be made to ensure support for such Scheduling. The project aims at designing and simulating a component of a proposed Distributed Scheduler, which schedules the client requests based on a COST Function. We intend to compare and contrast the COST based scheduling environment with a PRIORITY based and a FirstComeFirstServe (FCFS) scheduling scheme. Based upon results, guidelines for further research in the area of Distributed Scheduling will be provided.

1.Introduction

The core idea for the project has been taken from the Thesis submitted by Prof. Mitchell Theys [The99]. The simulations done in the thesis are static in nature. As we have all experienced, networks are not static, and this research intends to add dynamism to the model With the new model, we then apply some of the functions defined to the dynamic network model and try to simulate the network behavior under COST, PRIORITY[1] and FCFS Scheduling[2] schemes.

In any Network, there will always be contentions for the available network resources. Every user or client wants his request to be serviced ahead of the rest. In present systems, the user just sends in his request and waits for it to be satisfied. It is up to the Scheduler [GoH99] to ensure that the requests are satisfied.

[1] Scheduling based only on Priority
[2] Scheduling based on First Come First Served.

If the request is a normal request with no constraints of Deadline, the job of the scheduler is pretty simple. It just needs to distribute the client requests to appropriate File Servers. This has been the networking environment till early-middle 90's. But with support for Multi Media applications the situation has changed rapidly. Also changed is the way the Client requests are made. Quality of Service (QoS) [GaF98] began playing a critical role in Network Research and Design. Priority, Deadline and several other constraints came into the picture. This made the task of scheduling more complicated. As more and more constraints get added the time taken by the scheduler keeps increasing and this causes a bottleneck.

Scheduling in the early 90's used to be more centralized where the entire task of scheduling the Clients is done at a single point for a given Network. This has the draw back of the Scheduler getting blocked when the Network gets over subscribed and also the process time used to be very high.

Centralized Schedulers slowly began to pave the path for Distributed Schedulers and research has been focused on shifting the bottleneck to the end points from the center of a networking model. In addition to the Schedulers, Data Staging [The99] also became a critical factor in the design of the Dynamic Network model. The location of the Data Files with respect to the Clients and also the possibility that the same Data Item can be located at more than one location also adds more complexity. Data Staging has been considered in our research and it will play a significant role when the complete model is designed and simulated.

This project aims at simulating a subnet of a complete Distributed Scheduling environment and testing the behavior of the subnet using a COST based Scheduling mechanism and comparing the performance with that of a normal FCFS Scheduling mechanism and a PRIORITY scheduling mechanism. The role played by the Server module in our project is like a File Server or Data Source. By pushing the scheduling mechanism to this module we intend to reduce the bottleneck at the intermediate routers. The router [Hui00] transfers the Client request to appropriate File Servers based on the File name requested by the Client and the corresponding File Server is responsible for scheduling and servicing the Client requests. This ensures that none of the clients block the intermediate router. One new factor we propose in our project is File Versioning System [ThT00]. The idea is to test the adaptability of the model. There are multiple versions of the file and we intend to abstract this information from the client. Since our assumption is that the Server services the clients one at a time, when the Scheduler module returns an ordered list of clients based on COST then the Service handler services them in that order, one at a time. The Service handler decides which file version to be sent to the client. File versioning ensures that the client gets some version of the file within its deadline. This plays an important role in defense environments where, if the client request for a map of the target location, and the Server calculates that the request cannot be met within the deadline by using a higher resolution map, then it can send a lower resolution image that will meet the client's requirements. We intend to test this adaptability in our simulation.

One more advantage of having this type of Scheduler is that the Server knows ahead of time when each client is going to be serviced. Since we are dealing with hard Deadlines [The99], it is highly advantageous to inform the clients ahead of time if their request can be satisfied or not The Client need not wait till his turn to see if it can be

information about the valid values for the parameters is presented in Section 4.1.
These are assumptions specific to the simulation:

- Four different Files, each with three different versions.
- Three Priority levels are defined, HIGH, MEDIUM and LOW. Every Client is assigned one of these three Priority levels.
- The three Priority levels are equally distributed between the clients.
- Three Bandwidth values are defined, HIGH, MEDIUM and LOW. Every Client is assigned one of these three BandWidth values.
- The three Bandwidth values are equally distributed between the clients.
- Three Deadline values are defined. Every client is assigned one of these three DeadLine values.
- The three Deadline values are equally distributed between the clients.
- Every Client requests a Filename.
- The Four Data Files are equally distributed between the clients.
- The server considering Priority and Urgency parameters assign Cost based on the Cost function.

1.2 Terminology

Before discussing the design in more detail, some terms should be defined. These terms are used throughout the design and for clarity we present our definitions at this point.

Priority: This parameter specifies the Importance a Client expects from the Server. A value of HIGH implies the client expects it to be serviced ahead of rest.

DeadLine: This parameter specifies the maximum time a client can wait for its request to be serviced. It is the time before which a client expects its requested data.

BandWidth: This parameter specifies the capacity of the link between Client and the Server.

Urgency: The closeness of the Actual Transfer Time to the Deadline gives the urgency factor.

Cost: This parameter specifies the importance given to the client by the server. Server assigns this value based on a Cost Function.

SuccessClients: The list of clients that can be serviced within the specified deadline.

FailureClients: The list of clients that cannot be serviced within the specified deadline.

Transfer Time: It is the time taken by the file to get transmitted over the given bandwidth.

ActualTransferTime: The actual time taken for the file transfer. It includes Transfer Time + the ActualTransferTime of clients already serviced.

2. Design Specifications

2.1 Design

The design in this document refers to one subnet of a full-scale distributed network model that can be build by expanding this concept to multiple nodes. Server plays the central role in this design as shown in Fig. 1. All the clients get connected to this server, which also acts as the repository for the data files. The Server is capable of accepting multiple clients at a given time and it doesn't get blocked when a client waits long time for a data item. Figure 2 shows the internal modules of the server.

Figure 1 A Simple Server client setup

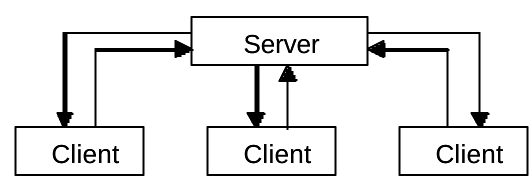
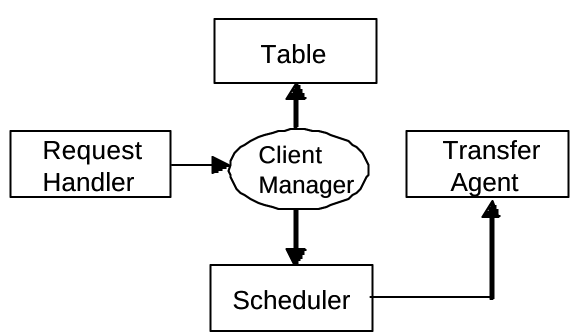


Figure 2 Internal modules of the Server



2.2 Internal working of the Server

2.2.1 Request Handler

This is responsible for handling client requests. When the client connects to the server port, the request handler may accept or reject the client connection. This decision is done based on the maximum limit on number of simultaneous client sessions that can be supported by the server.

2.2.2 Client Manager

Once the connection with the client is established, the client has to furnish some information pertaining to the connection to the Server as per the protocol agreed upon by the Client and the Server. The Client Manger assigns Cost to each client based on the Cost Function. The Client Manager stores all these information about every client for future reference.

2.2.3 Scheduler

Once the Client Manager finishes its task, control goes to the scheduler. The scheduler as the name suggests schedules the clients. Cost associated with the client is considered in scheduling the clients.

2.2.4 Transfer Agent

This enables the date transfer between the Clients and the Server in the order of clients returned by the scheduler function.

3. Implementation Details

The simulation has been done using Socket programming in Java [HuS99] using Multi Threading [OaW99] concepts. Multi Threading concept is used so that the Server doesn't get blocked by a single client and also to make one client independent of one other. Effort is to simulate a real world environment where one user session is independent of other. TCP/IP sockets are used to simulate the Server and the Client.

The Server opens up the "Server Socket" on a well-defined port and waits for Client connections. Simulation is done taking a specific number of clients at a time. When the client program is started, it spawns the appropriate number of client threads. Each Client thread establishes a connection with the server. An initialization is then performed that exchanges information as per the protocol. Next, the Scheduler assigns a Cost to each of the Clients based on the Cost Function. The Cost Function includes a balancing factor to give equal importance to both Priority and Deadline. Based on the Priority requested by the Client it is assigned a weight and same is the case with the urgency factor. We took the Inverse of the Normalization to be our balancing factor. Based on this cost function the clients are scheduled. The scheduler function returns two ordered lists of Clients, Success Clients and Failure clients. A client is defined to be a Success Client if it can get its required file before the deadline and is defined as Failure client if it fails to get the file within the deadline. Since our assumption is that only one client gets serviced at a time, we use a look ahead scheduler to know before hand whether a dient is success or fail. If it is a failure then the client is send an appropriate message and the connection is terminated. Because the network is over subscribed with requests and all the requests cannot be met within their deadline requirements, some scheduling mechanism is necessary to assure that a majority of the requests get satisfied.

Scheduling logic is applied for COST, PRIORITY and FCFS scenarios and the simulations are done 40 times and the results are tabulated.

4. Implementation Specifications

4.1 Parameter values

The Following section provides information about the DeadLine, Priority and other constraints used in the simulation.

No. Of Clients: 50, 100 & 300
DeadLine parameters: 1250, 1750 & 2250 Sec.
Bandwidth parameters: 250, 500 & 1000 Kbps
The three Weighted Priority values: 1, 5 & 10
The three Weighted Urgency values: 1, 5 & 10.

Cost=((Wp/Np) + (Wu/Nu))[3]

The four File Names and the sizes of their versions:

	File1	File2	File3	File4
Version1(bytes)	18750	24450	15000	20000
Version2(bytes)	9768	16760	8750	13876
Version3(bytes)	4500	12800	6345	9756

5. Case Studies

5.1 Case Study1

Simulation has been done 40 times with 50 clients.

Before Scheduling:

Average number of PriorityI requests: 17
Average number of PriorityII requests: 18
Average number of PriorityIII requests: 15
m: Minimum Number of Clients
A: Average Number of Clients
M: Maximum Number of Clients
P1: Number of PriorityI Clients that get serviced.
P2: Number of PriorityII Clients that get serviced.
P3: Number of PriorityIII Clients that get serviced.
V1: Number of Clients that get Higher Version File.

Using Schedulers:

	COST			PRIORITY			FCFS		
	m	A	M	m	A	M	m	A	M
P1	13	17	21	13	17	21	5	10	14
P2	12	14	16	12	14	16	12	15	18
P3	13	14	21	13	14	21	11	15	21
V1	43	46	50	43	46	50	35	39	43

5.2 Case Study2

Simulation has been done 40 times with 100 clients.

Before Scheduling:

Average number of PriorityI requests: 36
Average number of PriorityII requests: 28
Average number of PriorityIII requests: 36

[3] Wp: Weighted Priority Np: Normalized Priority
Wu: Weighted Urgency Nu: Normalized Urgency

Using Schedulers:

	COST			PRIORITY			FCFS		
	m	A	M	m	A	M	m	A	M
P1	23	31	35	23	30	34	03	08	18
P2	07	10	14	06	08	11	09	14	17
P3	23	27	30	23	26	28	24	32	42
V1	61	65	68	59	63	67	44	56	66

5.3 Case Study 3

Simulation has been done 40 times with 300 clients.

Before Scheduling:

Average number of PriorityI requests: 99
Average number of PriorityII requests: 102
Average number of PriorityIII requests: 99

Using Schedulers:

	COST			PRIORITY			FCFS		
	m	A	M	m	A	M	m	A	M
P1	37	39	41	31	33	35	04	09	15
P2	08	10	12	06	07	10	11	15	18
P3	25	29	34	24	27	29	33	42	52
V1	71	75	78	61	65	70	57	63	72

5.4 Analysis

In the simulations care has been taken while considering parameters like File Size, DeadLine, BandWidth so that the values represent real time environment. The COST function has also been defined giving both Priority and Urgency Factors equal weightage.

Based on the network design and constraints assumed the network gets overloaded when the client count becomes 100 and beyond that the improvement factor in terms of number of higher priority clients getting satisfied is minimum. This is due to the fact that the server handles one client at a time and the assumed file sizes are such that there is always contention for the slot that gets serviced immediately. Also we assumed only 4 Data Files each with 3 different file versions which results in the scenario where on average each Data File is requested by 25 Clients (when number of clients is 100) and by 75 clients (when the number of clients is 300). So the success percentage is acceptable considering all these constraints of heavy request rate.

6. Conclusion

The simulation results have been compared between COST based Scheduling, PRIORITY based Scheduling and FCFS based Scheduling for the number of High Priority Clients that are successful and the number of Clients that get HigherVersionFile. It is observed that using COST or PRIORITY based scheduling we get more number of Higher Priority Clients to be successful than using FCFS. Also the number of clients that are serviced with a Higher Version File is also more in COST and PRIORITY based Scheduling than FCFS. Once the full-scale model is developed with more number of subnets and with the scheduling between subnets also playing an important role we can expect to see much variations in the simulations of COST and PRIORITY based Scheduling.

The results presented here show that using our COST based scheduling we achieved a better success rate than normal FCFS or PRIORITY scheduling. The results are in agreement with our predictions prior to the implementation.

7. Scope for Further Research

In every Research field there is always scope for future enhancements. The following are few of them:

- When the system is enlarged with multiple subnets we can have a fully functional distributed system. This requires proper definition of the protocols to be followed for intra subnet communications and load balancing.
- A group of near by subnets can be made a "Zone" so that the complexity involving message transfers and scheduling can be narrowed down to the zone level.
- Consider handling multiple file transfers at a time. Imagine a pipe that can service any number of clients till its capacity is reached.
- Have multiple Data Sources and try to arrange for file transfer from any one of them based on the proximity to the requesting client.
- Redefine the COST Function to include some more variants that might play a critical role once the complete system is build. Some possibilities might be the "Zone Factor" which indicates whether the data request arriving at a node is from the same zone or from a different zone.

8. References

[GoH99] Ashish Goel, Monika R Herzinger, Plotkin S, and Tardos E, *"Scheduling Data Requests in a Network"*, Symposium on Theory of Computing, 1999.

[GaF98] Gaines, G. and Festa, M, *"Resource Reservation Protocol. - A Survey of RSVP / QOS Implementations"*, RSVP Working Group, July 1998.

[Hui00] Christian Huitema, *"Routing .in the Internet"*, 2d ed. Prentice-Hall Publications, 2000, pp 2-3.

[HuS99] Mernil Hughes, Michael Schoffner, and Derek Hammer *"Java Network Programming"*, 2d ed. Manning Publications, May 1999, pp. 4-5.

[OaW99] Scott Oaks, Henry Wong, and Mike Loukides *"Java Threads"*, 2d ed. O'Reilly & Associates, Jan 1999, pp 4-5.

[The99] Mitchell D. Theys, " *An Investigation of Models and Heuristics for Scheduling Data Requests in a Distributed Communication Environment,"* PhD Thesis, School of Electrical and Computer Engineering, Purdue University, August 1999.

[ThT00] Mitchell D. Theys, Min Tan, Noah B. Beck, Howard Jay Siegel, and Michael Jurczyk, *"A mathematical model and Scheduling Heuristics for satisfying prioritized data request .in an oversubscribed communication Network,"* IEEE Transactions on Parallel and Distributed Systems, Vol 11, No. 9, September 2000, pp. 969-988.