



Signalling and Scheduling for Efficient Bulk Data Transfer in Circuit-switched Networks

by

Reinette Grobler

Submitted in fulfillment of the requirements
for the degree M.Sc Computer Science in the
Faculty of Natural & Agricultural Sciences

University of Pretoria

Pretoria

August 2000

Signalling and Scheduling for Efficient Bulk Data Transfer in Circuit-switched Networks

by Reinette Grobler

Supervisor: Prof. D.G. Kourie

Co-supervisors: Prof. M. Veeraraghavan and Prof. J. Roos

Department of Computer Science

M.Sc Computer Science

Summary

The number and size of files transferred over the Internet is sufficiently significant to warrant the investigation of faster and more efficient mechanisms for their transfer. This research begins by comparing the performance of circuit-switching and TCP/IP (which is most commonly used in the Internet) when bulk data is transferred. Experimental and analytical analysis suggests that circuit-switching is indeed better suited for bulk data transfer than TCP/IP.

Circuit-switching suffers from the drawback that connections at rates higher than DS0 (64Kbps) can only be obtained in provisioned mode. For circuit-switching to be usable for bulk data transfer, it should support on-demand circuits at much higher data rates, for example OC1 (51.84Mbps) to OC768 (40 Gbps). A signalling protocol has been designed to be implemented in hardware (thus providing high throughput). It enables the setup of high-bandwidth on-demand circuits in Time Division Multiplexing (TDM) networks. An accompanying routing protocol to be implemented in software is also designed. Further, design decisions are considered for a transport protocol that enables the reliable transfer of bulk data over a circuit-switched connection.

When transferring bulk data over a connection-oriented network that implements preventative congestion control mechanisms, the connection has a deterministic holding time that can be computed from the file size, data transfer rate and propagation delays.

With the knowledge of connection duration it is possible to move away from the current blocking mode of operation (where a call is blocked when a switch does not have sufficient resources to handle it) and to implement a queueing mode of operation where connections are scheduled. By scheduling connections it is possible to reply to a connection request with a later start time if the required resources are not available at the time of the request.

Two scheduling schemes have been designed. Simulations show that at 70% loading, these schemes offer start time delays that are up to 85% smaller and channel utilization of up to 37% larger than a simple queueing mode of operation where call holding times are ignored when connections are set up.

Signalling and Scheduling for Efficient Bulk Data Transfer in Circuit-switched Networks

by Reinette Grobler

Supervisor: Prof. D.G. Kourie

Co-supervisors: Prof. M. Veeraraghavan and Prof. J. Roos

Department of Computer Science

M.Sc Computer Science

Opsomming

Die hoeveelheid en grootte van lêers wat tans oor die Internet versend word is noemenswaardig. Om hierdie rede is 'n ondersoek geloods op soek na meer doeltreffende meganismes vir die versending van massa data. Hierdie navorsing begin deur die werkverrigting van *circuit-switching* te vergelyk met die van TCP/IP (wat tans die mees algemene protokol is vir data versending oor die Internet). Eksperimentele en analitiese ondersoeke suggereer dat *circuit-switching* wel meer geskik is vir massa data versending as TCP/IP.

In *circuit-switched* netwerke is konneksies van slegs DS0(64Kbps) op aanvraag beskikbaar. All konneksies hoër as DS0 moet voorlopig bespreek word en word gewoonlik opgestel met die hulp van menslike interaksie. Vir *circuit-switching* om 'n noemenswaardige invloed te hê op massa data versending is daar 'n vereiste dat die netwerke konneksies teen baie hoër snelhede op aanvraag opgestel kan word. Byvoorbeeld, konneksies van OC1(51.84Mbps) tot OC768(40Gbps) gaan 'n baie groter invloed hê op die spoed van massa data versending as DS0 konneksies. 'n Protokol om konneksies op te stel (*signalling protocol*) is ontwerp om in apparatuur ge-implementeer te word. Deur die protokol in apparatuur te implementeer word die hoeveelheid konneksies wat per eenheid tyd opgestel kan word meer as wanneer 'n protokol gebruik word wat in programmatuur implementeer is. Hierdie protokol maak dit moontlik om konneksies vanaf OC1 tot OC192 op aanvraag in 'n TDM netwerk op te stel. 'n Saamgaande roeteringsprotokol (*routing protocol*), geteiken vir programmatuur implementasie, is ook ontwerp. Verder is opsies oorweeg vir 'n transport protokol wat betroubare versending van massa data oor 'n *circuit-switched* konneksie sal waarborg.

Wanneer massa data versend word oor 'n konneksie-geörienteerde netwerk wat voorkomende meganismes vir kongestiebeheer implementeer, het die konneksie 'n deter-



ministiese tydsduur wat bereken kan word vanaf die hoeveelheid data, die spoed van die konneksie en die totale verdragings tussen die twee nodes wat data uitruil.

Met die kennis van die tydsduur van die konneksie is dit moontlik om weg te beweeg van die huidige hantering van konneksies waar 'n konneksie geblok word indien daar op enige stadium van opstelling nie genoeg hulpbronne is om die konneksie te kan hanteer nie. Dit is nou moontlik om konneksies te skeduleer. Deur konneksies te skeduleer is dit moontlik om 'n versoek vir 'n konneksie te beantwoord met 'n later begintyd indien daar nie genoeg hulpbronne beskikbaar is wanneer die versoek arriveer nie.

Twee skeduleringskemas is ontwerp. Simulasies van die skemas wys dat teen 70% lading, die skemas vertraging in begintyd van konneksies verlaag met tot 85% en die benutting van kanale verhoog met tot 37% wanneer die skemas vergelyk word met 'n eenvoudige skema waar tydsduur ignoreer word tydens konneksie opstelling.



Acknowledgements

My sincere thanks to:

- my father, mother and brother for their continuous support throughout my studies;
- Prof. J. Roos for introducing me to the exciting world of computer networks;
- Prof. M. Veeraraghavan for her generous help and the amazing opportunities she presented to me;
- Prof. D.G. Kourie for his very helpful input;
- the techteam at the Computer Science Department of the University of Pretoria, Vafa Izadinia, Graeme Pyle, Edwin Peer, Jacques van Greunen and Eric Clements for all their technical assistance.



Contents

1	Introduction	1
2	Analyzing file transfer delays in circuit-switched and TCP/IP networks	6
2.1	Introduction	6
2.2	Comparing TCP/IP to circuit-switched networks	7
2.2.1	Circuit-switched networks	8
2.2.2	TCP/IP	10
2.3	Laboratory experiment	12
2.4	Numerical results	14
2.4.1	TCP/IP	14
2.4.2	UDP as an emulator of the circuit switched mode	16
2.4.3	The experimental results	16
2.4.4	The analytical results	18
2.5	Conclusion	23
3	Signalling protocol	25
3.1	Introduction	25
3.2	Implementing the signalling protocol in hardware	26
3.3	Mode of transport for signalling messages	28
3.4	Signalling protocol description	33
3.5	Signalling protocol specification	37
3.5.1	Messages	37
3.5.2	Description of Information elements	39
3.5.3	Tables	41
3.6	Setup and release procedures for homogeneous networks	43
3.6.1	Connection request	43

3.6.2	Connection establishment	44
3.6.3	Clearing a connection	44
3.7	Error conditions	47
3.7.1	Cause values	48
3.7.2	Timers	48
3.7.3	Handling of error conditions	50
3.8	Maximum bandwidth selection	52
3.9	Improvements to signalling protocol	53
3.9.1	Routing through heterogenous nodes	53
3.9.2	Transport of signalling messages	56
3.10	Conclusion	57
4	Routing protocol	58
4.1	Introduction	58
4.2	Addressing	59
4.3	Design decisions	62
4.4	Protocol description	65
4.4.1	Routing database and directed graph	66
4.4.2	Constructing the shortest path tree	70
4.4.3	Routing table	72
4.5	Future work	75
5	Scheduling calls with known holding times	78
5.1	Introduction	78
5.2	Motivating applications	80
5.3	Proposed algorithm	83
5.4	Extensions	91
5.4.1	Request for start time included in connection request	91
5.4.2	More than one switch in the network	91
5.5	Connection setup schemes	93
5.5.1	The kT_{wait} scheme	94
5.5.2	The $kT_{wait} - T_{max}$ scheme	95
5.5.3	The F scheme	95
5.5.4	The <i>timeslots</i> scheme	101



5.6	Simulation and results	104
5.6.1	Network model	104
5.6.2	Results	105
5.7	Further extensions and future work	112
5.7.1	Switch programming delay	112
5.7.2	Propagation delay computation	113
5.7.3	Time	114
5.8	Summary and Conclusions	114
6	Further work: The transport layer	116
6.1	Introduction	116
6.2	Protocol requirements	117
6.3	Conclusion	120
7	Conclusion	121
	Bibliography	123

List of Tables

1.1	Classification of networking techniques	1
1.2	Classification of data transfers	2
2.1	Values used for file sizes (f) in comparison of circuit-switching and TCP/IP	17
2.2	Parameters used in analytical comparison of circuit-switching and TCP . . .	19
3.1	Description of parameters in SEND primitive [10]	30
3.2	Description of parameters in RECV primitive [10]	31
3.3	Routing table	34
3.4	Available bandwidth table including cost	35
3.5	Connectivity table	36
3.6	Switch mapping table	37
3.7	SETUP	38
3.8	SETUP SUCCESS	38
3.9	RELEASE	38
3.10	RELEASE CONFIRM	39
3.11	Message types	39
3.12	State table	42
3.13	Cause classes	48
3.14	Cause values	49
3.15	Routing table of switch 3	55
4.1	ip_S to ip_R mapping table kept by SONET switch 1	62
4.2	Available bandwidth table including cost	66
4.3	Routing database entry for node D from network depicted in Figure 4.2 . . .	67
4.4	Routing update message	69
4.5	Possible values for the cross connect rate of a switch	70



4.6	Directed graph	70
4.7	Predecessor list for SwitchB	72
4.8	Routing table for node B	75
4.9	Routing table for Switch3 (OC1 and OC3 rates cannot be used)	76
5.1	Classification of data transfers	81
5.2	Different loads introduced by interference traffic	106
5.3	Parameters used in simulation of connection setup schemes	106

List of Figures

1.1	Chapter layout	5
2.1	Emission delays, TCP/IP vs. circuit switching	8
2.2	Network configuration used when transferring file with TCP/IP	13
2.3	Network configuration used when transferring file with UDP	13
2.4	Establishing a TCP connection	14
2.5	Closing a TCP connection	15
2.6	Comparison of transport protocols TCP and UDP	17
2.7	Effect of propagation delay on TCP and circuit switching	21
2.8	Effect of propagation delay on TCP and circuit switching, smaller files . . .	21
2.9	Effect of emission delay on TCP and circuit switching	22
2.10	Effect of emission delay on TCP and circuit switching, smaller files	22
3.1	Ideal network in which signalling protocol will be used	28
3.2	Typical network in which signalling protocol will be used	29
3.3	Primitives used between signalling process and IP module	29
3.4	Connectivity information	37
3.5	Signalling protocol state transition diagram	43
3.6	preceding/succeeding	43
3.7	Successful setup of a connection	45
3.8	Unsuccessful setup of a connection	45
3.9	Normal connection release	46
3.10	Connection release by an intermediate node	47
3.11	Connection setup with minimum and maximum bandwidth requirements . .	52
3.12	Connection setup in heterogenous network	54
3.13	Connection setup in large heterogenous network	55



4.1	Routing network	59
4.2	Example network configuration for routing protocol	66
4.3	Shortest path tree constructed by node B	72
4.4	Shortest path tree constructed by node B, with extra node	73
4.5	Shortest path tree constructed by Switch3	76
5.1	A simple one switch network	84
5.2	Block diagram of a switch	84
5.3	Keys used in flow diagrams	85
5.4	Flow chart for the message handling module in a one switch network	86
5.5	Flow chart for the message handling module in a one switch network, with error handling	87
5.6	Route lookup process included in the routing module	88
5.7	Process to program the switch fabric in the switch programming module	88
5.8	Scheduling process included in the CAC module for a one switch network	89
5.9	Process in the CAC module to undo the changes made by the scheduling process	90
5.10	Scheduling process (allowing for a host to request a start time) included in the CAC module for a one switch network	92
5.11	Successful setup of a connection	93
5.12	Connection setup at the ingress switch of a connection (F scheme)	98
5.13	Process in the CAC module to update the changes made by the scheduling process in the reverse direction	99
5.14	Example of scheduled connection setup	100
5.15	Available bandwidth of Switch1, interface i	101
5.16	Available bandwidth of Switch2, interface i	103
5.17	Network model	105
5.18	Percentage of calls blocked using different connection setup schemes	107
5.19	Percentage of calls blocked using different connection setup schemes, constant mean interarrival times	108
5.20	Queueing delay experienced by connection requests for connections requiring channel Switch1 -> Switch2	109
5.21	Start time delays for all connection setup schemes	109



5.22 Start time delays for all connection setup schemes, constant mean interarrival times 111

5.23 Utilization of channel Switch1->Switch2 111

5.24 Utilization of channel Switch1->Switch2, constant mean interarrival times . 112

6.1 Transport layer in a connection oriented circuit switched network 118



Acronyms

ATM Asynchronous Transfer Mode

CAC Connection Admission Control

CL Connectionless

CO Connection oriented

ICMP Internet Control Message Protocol

IETF Internet Engineering Task Force

IP Internet Protocol

OSPF Open Shortest Path First

PDH Plesiochronous Digital Hierarchy

PSTN Public Switched Telephone Network

QoS Quality of Service

RIP Routing Information Protocol

SCTP Stream Control Transmission Protocol

SDH Synchronous Digital Hierarchy

SONET Synchronous Optical Network

TCP Transmission Control Protocol

TDM Time Division Multiplexing

TOS Type of Service



TTL Time To Live

UDP User Datagram Protocol

WDM Wavelength Division Multiplexing