Middle-East Journal of Scientific Research 24 (Techniques and Algorithms in Emerging Technologies): 156-161, 2016
ISSN 1990-9233;

DOI: 10.5829/idosi.mejsr.2016.24.TAET23421

Flow Control With Data Compression To Avoid Overload In Networks

C. Vigundhiya and E. Menaka

Department of Computer Science and Engineering,
Vivekananda Institute of Engineering and Technology for Women, Tiruchengode, Tamilnadu, India

Abstract: Utility maximization in networks where the sources do not employ flow control and may consequently overload the network. In the deficiency of flow control at the sources, some packets will without doubt have to be dropped when the network is in overload. To that end, first develop a spread threshold based packet-dropping policy that maximizes the weighted sum throughput. Next, consider utility maximization and develop a receiver-based flow control scheme that, when joint with threshold-based packet dropping, achieves the optimal utility. The flow control scheme creates virtual queues at the receiver as a push-back mechanism to optimize the amount of data deliver to the destinations via back-pressure routing. Their near-optimal performance is proved and further supported by simulation results. In this paper we proposed by People tend to store a lot of files inside theirs storage space. When the storage nears it limit, they then try to reduce those files size to minimum by using data compression software. This algorithm will manipulate each bit of data inside file to reduce the amount without lose any data after decode which is classified to lossless compression.

Key words: Finite-Buffer Networks · Flow Control · Network Overload · Queuing Analysis · Utility Maximization

INTRODUCTION

Networking is the construction, design and use of a network, including the physical (cabling, hub, bridge, switch, router and so forth), the choice and use of telecommunication protocol [1] and computer software for using and managing the network and the organization of operation policies and events related to the network. Flows are categorize into classes so that flows in a class have a shared destination. A class may simply be a flow particular by a source–destination pair, or corresponds to a sub set of flows that communicate with a common Website. A utility purpose is assigned to each traffic class and the sum of the class-based utilities is maximize as a means to control the collective throughput of flows in each class. The use of class-based utility function is partly motivated by the need of justifying network congestion [2] caused by a collection of data flows whose combined through put needs to be controlled. Without flow control at the sources, some packet Will be dropped when the network is overloaded. To provide differentiate services to multiple traffic classes, believe the Scenario where the destination scan perform flow control to regulate the established throughput of each traffic class. The question to answer is how to design in-network packet dropping And receiver-based flow control strategies to maximize the sum Of class-based utilities and stabilize the network. In-network packet dropping and receiver-based flow control [3] enhance the robustness of network operations.

Overloading: When a single Object has multiple behaviors. Then it is called as Overloading. congestion is that in which a Single Object has a same name and Provides Many Functions. In Overloading followings belongings denotes Overloading:

- When an Object has Same Name.
- Difference is Return type.
- Difference in meaning, with Multiple Arguments.

Different Types of Data Overloading

Constructor Overloading: Constructor overloading is that in which a Constructor has a same name and has lots of Functions, then it is called as Constructor Overloading. As we Know that Constructor are of non-payment

Corresponding Author: C. Vigundhiya, Department of Computer Science and Engineering,
Vivekananda Institute of Engineering and Technology for Women, Tiruchengode, Tamilnadu, India.

Parameterized and Copy Constructors. So that when we are make a Single Constructor with various Arguments then it is called as Constructor Overloading.

Operator Overloading: Operators are used for Performing Operations on the Operands. But Each and Every Operator has Some borders resources an Operator which is also called as double are used for Performing Operations on the two Operands and Unary Operators perform their Operation on the single Operand. So with the help of Operator Overloading, we can Change the process of the Operator. Means With the help of Operators we can Change the Operation of the operator. For Example with the help of Binary Operators we can add two Objects Means not only the two Data Members of the Class.

Method Overloading: Method Overloading is also called as Function Overloading. Overloading Means a Functions has many Behaviors occurred When in class when a functions has same name but different behaviors. A Functions said to be filled to capacity When: - In Overloading followings things denotes Overloading:-

- When an Object has Same Name.
- Difference is Return type.
- Difference in Function, with Multiple Arguments.
- Difference in Data Type.

Network Model: Backpressure routing refers to an algorithm for dynamically routing traffic [4] over a multi-hop network by using congestion gradient. It usually refers to a data network, but can apply to other types of networks as well. Focus on the data network application, where multiple data streams arrive to a network and must be delivered to appropriate destination. This is similar to how water would flow through a network of pipes via pressure gradient. However, the backpressure algorithm can be applied to multi-commodity networks and to networks where transmission rates can be chosen from different (possibly time-varying) options. Attractive features of the backpressure algorithm are:

- it leads to utmost network throughput,
- Provably robust to time-varying network conditions,
- Realized without knowing traffic arrival rates or direct state probabilities.

Non-persistent TCP connections in transient overload situation, under the supposition that all connections have the same round-trip times. Goal is to develop theoretical tools that will allow us to relax this statement and obtain explicit expressions for the rate of growth of the number of connections at the system, the rate at which TCP connections leave the system, as well as the time needed for the completion of a connection. To that end, the system as a DPS (Discriminatory Processor Sharing) system which we analyze under very mild assumption on the likelihood distributions related to different classes of arrivals: only assume that the arrival rates of connections exist and that the amount of in order transmitted during a connection of a given type forms a inactive ergodic sequence.

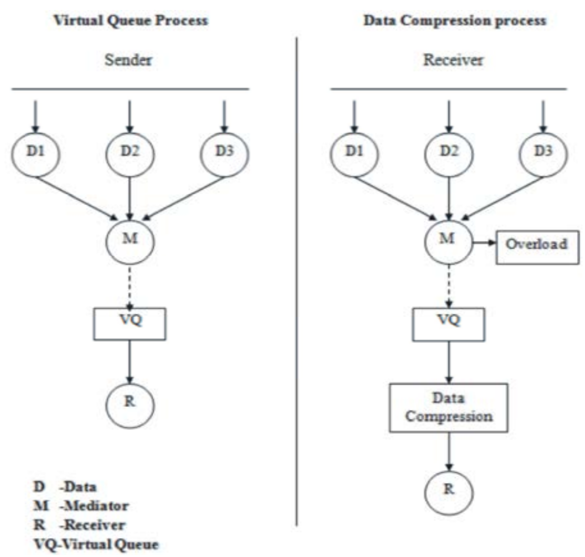

Fig. 2.1: Virtual Route Creation With Data Compression

The threshold-based packet-dropping policy, without the use of flow control, suffices to exploit the weighted sum throughput. furthermore, the combined flow control and packet-dropping mechanism has the following properties.

- It is distributed and only requires information alternate between neighboring nodes.
- It uses finite-size buffers.
- It is nearly utility-optimal (throughput-optimal as a special case) and the performance gap from the optimal utility goes to zero as buffer sizes increase.
- It does not need the information of arrival rates and thus is robust to time-varying coming rates that can go far beyond the network's stability region.
- This policy works effortlessly without the need of openly deciding whether a network enters or leaves an overload period.

- The policy can be implementing in parts of a network that include the receivers, treating the rest of the network as exogenous data sources.

Flow Control Process: Flow control is the process of adjust the flow of data from one device to another or between nodes on a network to ensure that the in receipt of device or node can handle all of the incoming data.

This is mainly important where the sending device or node is accomplished of sending data much faster than the receiving device or node can receive it. Too much data incoming before a device or node can handle it causes data overflow, resulting in data loss and possible retransmission. Flow of control is the order in which a program make actions.

Multiple File Path Allocation: Flow control in data networks aims to give fair allocation of resources [5] and regulate the source duty of traffic flows in order to stop network overload. In recent years, network effectiveness maximization problems have been studied to optimize network performance from side to side a arrangement of flow control, routing and scheduling, whose optimal operations are exposed as the solution to the utility maximization problems.

Queuing Analysis: The simplest queuing system [6] is depicted. The server is the central element of the system, which provides some service to items. Items from some residents of items arrive at the system to be served. If the server is idle, an item is served immediately. Otherwise, an incoming item joins a to come line2. The item gets depart when the server completes serving an item. If it waits in the queue, one among them is dispatched immediately to the server. The server in this model can represent everything that performs some function or service for a collection of items. Examples: a processor provide service to processes; a transmission line provides a transmission service to packet or frame of data; an I/O device provides a read or write service for I/O requests.

Finite-buffer Overflow: The set of realizable throughput vectors in terms of queue overflow rates. Third, using a simple breakdown of the utility functions, we design a network control policy [7] consisting of:

- a set of flow controller at the receivers;
- a packet-dropping device at internal nodes; and
- back-pressure routing between intermediate nodes.

The receiver-based flow controllers adjust throughput by modifying the differential backlogs between the receivers and their neighboring nodes-a small (or negative) degree of difference backlog is regarded as a push-back mechanism to slow down data delivery to the receivers. To deal with undeliverable data due to network overload, we design a threshold-based packet-dropping mechanism that discards data whenever queues grow beyond convinced thresholds.

Virtual Queue Construct: A new virtual queue-based back-pressure development VBR, which pre-establishes gradient at each node in a WSN and integrates this gradient when calculating the queue backlog differential between neighboring nodes when making back-pressure based scheduling [8] decision. The throughput optimality of VBR. Simulation results show that VBR significantly improves the network performance in packet delivery ratio, average E2E delay and average queue length at each node as compared with existing work.

Utility-Optimal Overload Resilient Algorithm (UORA): The data packet loss in the data transaction, the proposed method that is Utility-Optimal Overload Resilient Algorithm(UORA) create a virtual Queue. The large size of file will more from normal data queue to virtual queue and the operation process will continue as a parallel. So the propose methods execute and send the file very quickly to the receiver. The backpressure algorithm operate in slotted time and every slot it seeks to route data in instructions that maximize the differential backlog between neighboring nodes. In-network packet dropping and receiver-based flow control enhance the robustness of network operations.

Utility Maximization: To improving buffer queue by sending compressed packets throw this way we can increase the buffer space and maximize the utility of network. And improve the speed of network The packet compression technique groups packet of the same type and reads them using the double transposed feed, which experiences far less transitions than the traditional feed. This allows to use very simple and efficient compression algorithms (e.g. Run Length Encoding) to achieve good compression rates. The result is so simple that can be easily implemented on-board.

Virtual Queue Create: Normal Queue sender sends the data to receiver by mediator. Whereas in virtual queue sender sends the data to the mediator where the data is overloaded in mediator. Hence virtual queue used to decrease the overload that is occurred in mediator from virtual queue the is been send to the receiver. The data packet loss during the data transaction, the proposed method that is Utility-Optimal Overload Resilient Algorithm (UORA) create a virtual Queue.

The large size of file will more from normal data queue to virtual queue and the transaction process will continue as a parallel. So the propose method perform and send the file very quickly to the receiver. A virtual router can replace a physical router by using your computer and a software list called Connectives. Connectify works with Windows 7 computer and a exact Wife card chipset. When using a virtual router, you still make a network name but there isn't an extra matter piece of hardware to carry around. The virtual router is your mainframe it is the 'hotspot'.

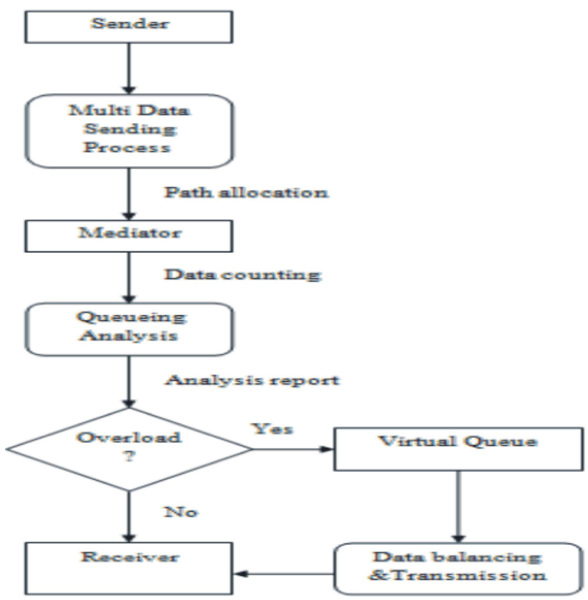

Fig. 2.2: Virtual Route Creation

Compression Techniques: Digitised sound and video produces a lot of data. In exacting digitised television quality pictures produce data at 270 Mbits/second which is faster than most hard disks, CD roms and network strategy can accommodate. We need to compress data for use on computers.

Compression is used just about all over the place. All the imagery you get on the web are compressed, typically in the JPEG or GIF formats, most modems use compression, HDTV will be compressed using MPEG-2 and several file systems automatically compress files when stored and the rest of us do it by hand. The neat thing about compression, as with the other topics we will cover in this course, is that the algorithms used in the real world make heavy use of a wide set of algorithmic tools, including sorting, hash tables, tries and FFTs. in addition algorithms with strong academic foundations play a dangerous role in real-world applications.

- Two types of compression.
- Lossless Compression Methods
- Lossy Compression Methods
- Lossless Compression Methods

Through lossless methods, original data and the data after compression and decompression are accurately the same. Lossless methods are used when we can't afford to lose any data: legal and medical documents, computer programs.

Run-length Encoding: RLE is a very simple form of data compression in sequence in which the same data value occurs in many consecutive data elements are stored as a single data value and count, rather than as the original run.

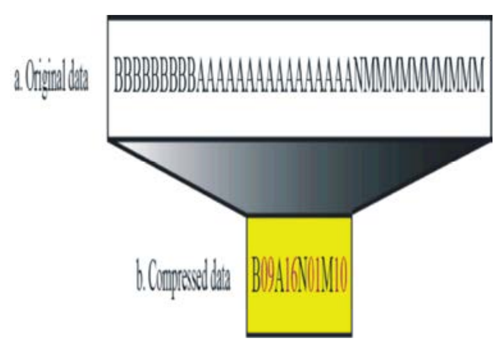

Fig. 2.3: Eg For Compression Data

This is useful on data that contains many such runs, for example, comparatively simple graphic images such as icons, line drawings, and animation. It is not useful with files that don't have lots of runs as it could potentially file size is increase

Huffman Coding: This is a general technique for coding secret code based on their arithmetical occurrence frequencies (probabilities). The pixels in the image are treat as symbols. The symbols that occur more commonly are assign a smaller number of bits, while the symbols that occur less regularly are assign a relatively larger number of bits.

Lempel Ziv Encoding: LZW (Lempel- Ziv – Welch) is a dictionary based coding. Dictionary based coding can be static or dynamic.

In static dictionary coding, vocabulary is fixed during the encoding and decoding process. In dynamic dictionary coding, the dictionary is efficient on fly. LZW is widely used in computer business and is implement as compress command on UNIX..

Lossy Compression Methods: Used for compressing images and video files (our eyes cannot tell apart subtle changes, so lossy data is acceptable). These methods are cheaper, less time and space.

Transformation Coding: This is the basis of achieving the compression. Only those few important coefficients are selected and the enduring are discarded.

The selected coefficients are considered for further quantization and entropy encoding. DCT coding has been the most common approach to convert coding. It is also adopted in the JPEG image compression standard.

Vector Quantization: A vocabulary of fixed-size vectors, called code vectors is developed. A vector is usually a block of pixel values. A given image is then partitioned into non-overlap blocks (vectors) called picture vectors.

Fractal Coding: The essential idea here is to decompose the image into segments by using standard image processing techniques such as color division, edge detection and spectrum and texture analysis. A library of fractals is used at each segment.

Block Truncation Coding: In this scheme, the image is divided into non overlapping blocks of pixels. For each block, threshold and restoration values are determined. The threshold is usually the mean of the pixel values in the block.

Sub Band Coding: In this scheme, the image is analyzed to produce the components contain frequencies in well-defined bands, the sub bands. after, quantization and coding is applied to each of the bands.

CONCLUSION AND FUTURE WORK

A receiver-based flow control and threshold based packet-dropping policy to cope with network overload and achieve optimal utility. Our format is robust to uncooperative users who do not employ source-end flow manage and to malicious users that intentionally overload the network A novel feature of our policy is a receiver-based backpressure/push-back device that regulates data flows at the granularity of traffic classes, where packets can be classified based on their types. This is in contrast to source-based scheme that can only differentiate between source–destination pairs. The receiver-based flow control scheme has a wide range of possible applications, including prevent denial-of-service attacks [9, 10] in Web servers, mitigating overload situation that may arise when the network is experience significant poverty due to a adversity or attack and even regulating traffic flows in the Internet. This framework also gives rise to a number of future research directions, such as accounting for the “cost” of packet dropping (e.g., due to the need to retransmit the dropped packets). A closely related problem involves the communication between TCP-based flow control and the receiver-based flow control scheme, e.g., TCP's response to the packet-dropping mechanism. In this context, it would also be motivating to develop a mathematical model to study optimal overload control in a network serving TCP flows. Another motivating future research direction is to use our framework to study traffic off-load problems in wire line and wireless networks, where traffic offloading is analogous to “dropping” data from the overloaded network to an option backup network. Data compression stills an important topic for research these days and has many applications and useful needed. the current data compression method might be far away from the ultimate limits, Run length algorithm provides important compression results, reducing arrangement size (and thus bandwidth requirements) by a factor of 3.60.

ACKNOWLEDGMENT

This research was supported by my Head of the Department, Prof.T.R.Srinivasan. We thank Mr.Chandra Mohan for assistance with a technique and Mr. Sathish for comments that greatly improved the manuscript. We thank famous persons Dr. Karunanithi and Dr. K.C.K. Vijayakumar, for sharing their pearls of wisdom with us during the course of this research and we thank Mrs. E. Menaka and Mr. R. Rajagopal reviewers for their so-called insights. We also immensely grateful to family for their comments and support, although any errors are our own and should not tarnish the reputations of these esteemed persons.

REFERENCES

1. Kelly, F.P., 1997. "Charging and rate control for elastic traffic," Eur. Trans. Telecommun., 8: 33-37.
2. Eryilmaz, A. and R. Srikant, 2006. "Joint congestion control, routing and MAC for stability and fairness in wireless networks," IEEE J. Sel. Areas Commun., 24(8): 1514-1524, Aug. 2006.
3. Low, S.H. and D.E. Lapsley, 1999. "Optimization flow control-I: Basic algorithm and convergence," IEEE/ACM Trans. Netw., 7(6): 861–874, Dec. 1999.
4. Kelly, F.P., A.K. Maulloo and D.K.H. Tan, 1998. "Rate control in communication networks: Shadow prices, proportional fairness and stability,"J. Oper. Res., 49: 237-252.
5. Eryilmaz, A. and R. Srikant, 2007. "Fair resource allocation in wireless networks using queue-length-based scheduling and congestion control, "IEEE/ACM Trans. Netw., 15(6): 1333-1344, Dec. 2007.
6. Stolyar, A.L., 2005. "Maximizing queueing network utility subject to stability: Greedy primal-dual algorithm," Queueing Syst., 50(4): 401-457.
7. Neely, M.J., E. Modiano and C.P. Li, 2008. "Fairness and optimal stochastic control for heterogeneous networks," IEEE/ACM Trans. Netw., 16(2): 396-409, Apr. 2008.
8. Lin, X. and N.B. Shroff, 2004. "Joint rate control and scheduling in multihop wireless networks," in Proc. IEEE CDC, Dec. 2004, pp: 1484-1489.
9. Srivastava, A., B.B. Gupta, A. Tyagi, A. Sharma and A. Mishra, 2011. "A recent survey on DDoS attacks and defense mechanisms," in Advances in Parallel Distributed Computing, ser. Communications in Computerand Information Science. Berlin, Germany: Springer, 203: 570-580.
10. Chang, R.K.C., 2002. "Defending against flooding-based distributed denialof-service attacks: A tutorial," IEEE Commun. Mag., 40(10): 42-51, Oct. 2002.