Volume 4, Issue 8, August 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Using Fuzzy Logic Intellect Spatial Control to Provide Traffic Managerial Service under High Speedy Networks

[1]V. P. Jayasudha, [2]Mr. D. Bulla Rao, [3]Mr. P. Nageswara Rao
[1]BTech, MBA., MTech II year (pursuing) Student, Swetha Institute of Technology & Science, Tirupati, India
[2]M.Tech, Assistant professor in dept of CSE, Swetha Institute of Technology & Science, Tirupati, India
[3]M.Tech, (Ph.D)., Associate professor in dept of CSE, Swetha Institute of Technology & Science, Tirupati, India

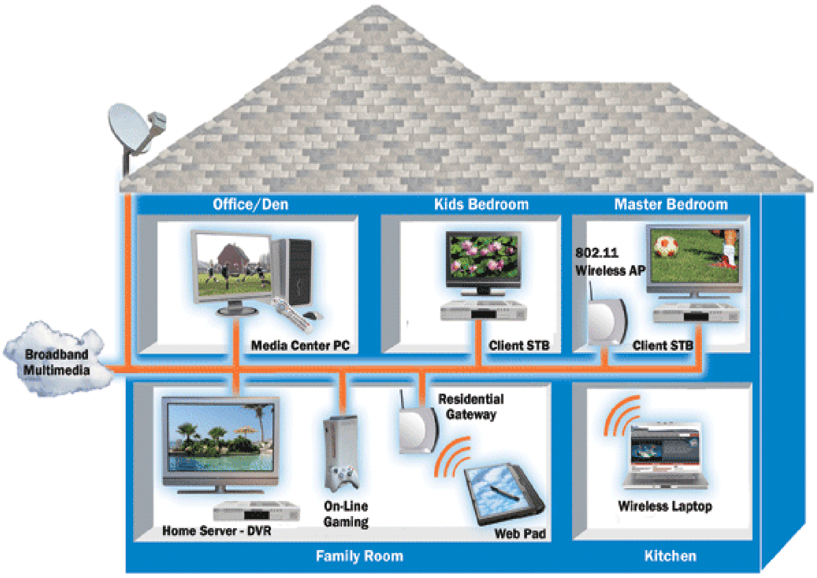
A class of explicit congestion congestion control protocols has been proposed to signal network traffic level more precisely by using multiple bits. In view of the fast-growing Internet traffic, this paper propose a distributed traffic management framework, in which routers are deployed with intelligent data rate controllers to tackle the traffic mass. Unlike other explicit traffic control protocols that have to estimate network parameters (e.g., link latency, bottleneck bandwidth, packet loss rate, or the number of flows) in order to compute the allowed source sending rate, our fuzzy-logic-based controller can measure the router queue size directly; hence it avoids various potential performance problems arising from parameter estimations while reducing much consumption of computation and memory resources in routers. The communication QoS (Quality of Service) is assured by the good performances of our scheme such as max-min fairness, low queueing delay and good robustness to network dynamics. Simulation results and comparisons have verified the effectiveness and showed that our new traffic management scheme can achieve better performances than the existing protocols that rely on the estimation of network parameters.

Keywords: congestion spatial control, fuzzy logic control, quality of service, traffic management, fast moving network, robustness.

I. INTRODUCTION

What is networking?

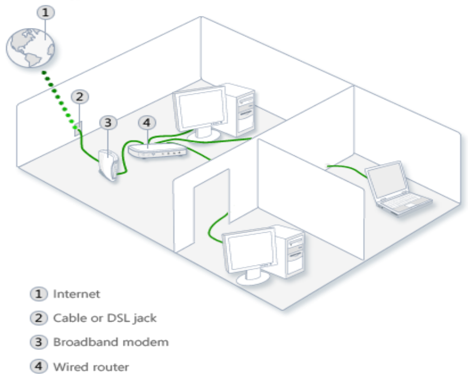
Networking is the word basically relating to computers and their connectivity. It is very often used in the world of computers and their use in different connections. The term networking implies the link between two or more computers and their devices, with the vital purpose of sharing the data stored in the computers, with each other. The networks between the computing devices are very common these days due to the launch of various hardware and computer software which aid in making the activity much more convenient to build and use.



Structure of Networking between the different computers

How networking works?

General Network Techniques - When computers communicate on a network, they send out data packets without knowing if anyone is listening. Computers in a network all have a connection to the network and that is called to be connected to a network bus. What one computer sends out will reach all the other computers on the localnetwork



Above diagrams show the clear idea about the networking functions

For the different computers to be able to distinguish between each other, every computer has a unique ID called MAC-address (Media Access Control Address). This address is not only unique on your network but unique for all devices that can be hooked up to a network. The MAC-address is tied to the hardware and has nothing to do with IP-addresses. Since all computers on the network receives everything that is sent out from all other computers the MAC-addresses is primarily used by the computers to filter out incoming network traffic that is addressed to the individual computer.
When a computer communicates with another computer on the network, it sends out both the other computers MAC-address and the MAC-address of its own. In that way the receiving computer will not only recognize that this packet is for me but also, who sent this data packet so a return response can be sent to the sender.2**On an Ethernet network** as described here, all computers hear all network traffic since they are connected to the same bus. This network structure is called multi-drop.
One problem with this network structure is that when you have, let say ten (10) computers on a network and they communicate frequently and due to that they sends out there data packets randomly, collisions occur when two or more computers sends data at the same time. When that happens data gets corrupted and has to be resent. On a network that is heavy loaded even the resent packets collide with other packets and have to be resent again. In reality this soon becomes a bandwidth problem. If several computers communicate with each other at high speed they may not be able to utilize more than 25% of the total network bandwidth since the rest of the bandwidth is used for resending previously corrupted packets. The way to minimize this problem is to use network switches.

II. FUZZY LOGIC SPATIAL CONTROL

The contributions of our work lie in: Using fuzzy logic theory to design an explicit rate-based traffic management scheme (called the IntelRate controller) for the high-speed IP networks; The application of such a fuzzy logic controller using less performance parameters while providing better performances than the existing explicit traffic control protocols; The design of a Fuzzy Smoother mechanism that can generate relatively smooth flow throughput; The capability of our algorithm to provide max-min fairness even under large network dynamics that usually render many existing controllers unstable.

An implementation and experimental study of the explicit control protocol

The explicit control protocol (XCP) has been proposed as a multi-level network feedback mechanism for congestion control of Internet transport protocols. Theoretical and simulation results have suggested that the protocol is stable and efficient over high bandwidth-delay product paths, while being more scalable to deploy than mechanisms that require per-flow state in routers. However, there is little operational experience with the approach. Since the deployment of XCP would require changes to both the end hosts and routers, it is important to study the implications of this new architecture before advocating such wide scale changes to Internets. This paper presents the results of an experimental study of XCP. We first implemented XCP in the Linux kernel and solved various systems issues. After validating previously reported simulation results, we studied the sensitivity of XCP's performance to various environmental factors, and discovered issues with TCP/IP configuration, capacity misestimation due to link sharing, handling of non-congestion losses, and the partial deployment of XCP queues in the network. These sensitivities can significantly reduce XCP's ability to control congestion and achieve fairness. Our contributions are twofold. First, through implementation we have revealed the challenges in platforms that lack large native data types or floating point arithmetic, and the need to keep fractions in the XCP protocol header. Second, through experiment and analysis we have identified several possibilities for XCP to enter into incorrect feedback control loops and adversely affect the performance. The challenges identified are deployment challenges intrinsic to the XCP design, and they suggest that the current proposal requires additional development and extension.

III. CONGESTION CONTROL

JetMax: scalable max-min congestion control for high-speed heterogeneous networks

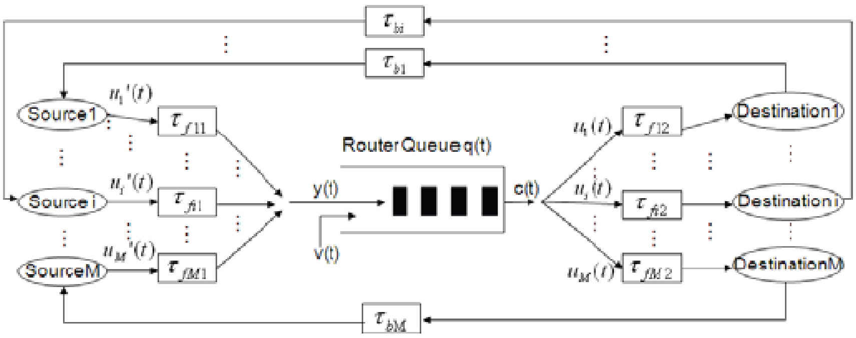
Recent surge of interest towards congestion control that relies on *single-link* feedback (e.g., XCP, RCP, MaxNet, EMKC, VCP), suggests that such systems may offer certain benefits over traditional models of additive packet loss. Besides topology-independent stability and faster convergence to efficiency/fairness, it was recently shown that any stable single-link system with a symmetric Jacobian tolerates arbitrary fixed, as well as *time-varying*, feedback delays. Although delay-independence is an appealing characteristic, the EMKC system developed in exhibits undesirable equilibrium properties and slow convergence behavior. To overcome these drawbacks, we propose a new method called JetMax and show that it admits a low-overhead implementation inside routers (three additions per packet), overshoot-free transient and steady state, tunable link utilization, and delay-insensitive flow dynamics. The proposed framework also provides capacity-independent convergence time, where fairness and utilization are reached in the same number of RTT steps for a link of *any* bandwidth. Given a 1 mb/s, 10 gb/s, or googol (10^{100}) bps link, the method converges to within 1% of the stationary state in six RTTs. We finish the paper by comparing JetMax's performance to that of existing methods in ns2 simulations and discussing its Linux implementation.

IV. TRAFFIC MANAGEMENT PRINCIPLE

Considering a backbone network interconnected by a number of geographically distributed routers, in which hosts are attached to the access routers which cooperate with the core routers to enable end-to-end communications. Congestion occurs when many flows traverse a router and cause its IQSize to exceed the buffer capacity, thus making it a bottleneck in the internet. Inside each router, our distributed traffic controller acts as a data rate regulator by measuring and monitoring the IQSize. As per its application, every host requests a sending rate it desires by depositing a value into a dedicated field Req_rate inside the packet header. The assumptions pertain are every source requests a desired sending rate from the network according to its application. A destination always has enough buffer space to receive data from its source. the requesting discipline of routers is FIFO. Long lived flows with infinitely long files are used to approximate the greedy behavior of a source when active.

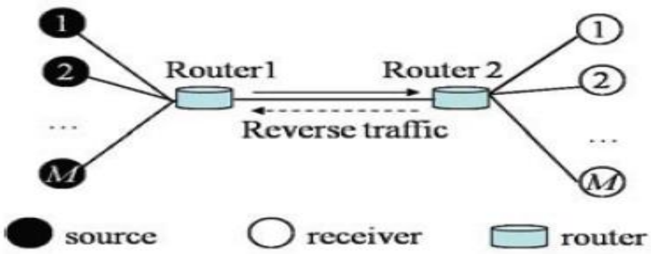
SYSTEM DESIGN

SYSTEM ARCHITECTURE:



x

SIMULATION SETUP:



V. CONCLUSION

A novel traffic management scheme, called the IntelRate controller, has been proposed to manage the Internet congestion in order to assure the quality of service for different service applications. The controller is designed by paying attention to the disadvantages as well as the advantages of the existing congestion control protocols. As a distributed operation in networks, the IntelRate controller uses the instantaneous queue size alone to effectively throttle the source sending rate with max-min fairness. Unlike the existing explicit traffic control protocols that potentially suffer from performance problems or high router resource consumption due to the estimation of the network parameters, the IntelRate controller can overcome those fundamental deficiencies. The reason of the zero packet loss is that the InterRate controller can always control the variations of the IQSize around the TBO position. To verify the effectiveness and superiority of the IntelRate controller, extensive experiments have been conducted in OPNET modeler. In addition to the feature of the FLC being able to intelligently tackle the nonlinearity of the traffic control systems, the success of the IntelRate controller is also attributed to the careful design of the fuzzy logic elements.

REFERENCES

[1] M. Welzl, *Network Congestion Control: Managing Internet Traffic*. John Wiley & Sons Ltd., 2005.

[2] R. Jain, "Congestion control and traffic management in ATM networks: recent advances and a survey," *Computer Networks ISDN Syst.*, vol. 28, no. 13, pp. 1723–1738, Oct. 1996.

[3] D. Katabi, M. Handley, and C. Rohrs, "Congestion control for high bandwidth-delay product networks," in *Proc. 2002 SIGCOMM*, pp. 89– 102.

[4] D. Katabi, M. Handley, and C. Rohrs, "Congestion control for high bandwidth-delay product networks," in *Proc. 2002 SIGCOMM*, pp. 89– 102.

[5] N. Dukkipati, N. McKeown, and A. G. Fraser, "RCP-AC congestion control to make flows complete quickly in any environment," in *Proc. 2006 IEEE INFOCOM*, pp. 1–5

[6] Y. Zhang, D. Leonard, and D. Loguinov, "JetMax: scalable max-min congestion control for high-speed heterogeneous networks," in *Proc. 2006 IEEE INFOCOM*, pp. 1–13.

[7] B. Wydrowski, L. Andrew, and M. Zukerman, "MaxNet: a congestion control architecture for scalable networks," *IEEE Commun. Lett.*, vol. 7, no. 10, pp. 511–513, Oct. 2003.

[8] Y. Zhang and M. Ahmed, "A control theoretic analysis of XCP," in *Proc. 2005 IEEE INFOCOM*, vol. 4, pp. 2831–2835.

[9] Y. Zhang and T. R. Henderson, "An implementation and experimental study of the explicit control protocol (XCP)," in *Proc. 2005 IEEE INFOCOM*, vol. 2, pp. 1037–1048.

[10] J. Pu and M. Hamdi, "Enhancements on router-assisted congestion control for wireless networks," *IEEE Trans. Wireless Commun.*, vol. 7, no. 6, pp. 2253–2260, June 2008.

[11] F. Abrantes, J. Araujo, and M. Ricardo, "Explicit congestion control algorithms for time varying capacity media," *IEEE Trans. Mobile Comput.*, vol. 10, no. 1, pp. 81–93, Jan. 2011.

[12] L. Benmohamed and S. M. Meerkov, "Feedback control of congestion in packet switching networks: the case of a single congested node," *IEEE/ACM Trans. Netw.*, vol. 1, no. 6, pp. 693–708, Dec. 1993.

[13] S. Chong, S. Lee, and S. Kang, "A simple, scalable, and stable explicit rate allocation algorithm for max-min flow control with minimum rate guarantee," *IEEE/ACM Trans. Netw.*, vol. 9, no. 3, pp. 322–335, June 2001.

[14] S. J. Lee and C. L. Hou, "A neural-fuzzy system for congestion control in ATM networks," *IEEE Trans. Syst. Man Cybern. B., Cybern.*, vol. 30, no. 1, pp. 2–9, 2000.

[15] A. Vashist, M. Siun-Chuon, A. Poylisher, *et al.*, "Leveraging social network for predicting demand and estimating available resources for communication network management," in *Proc. 2011 IEEE/IFIP Int. Symp. Integrated Netw. Manage,* pp. 547–554.

[16] D. Toelle and R. Knorr, "Congestion control for carrier ethernet using network potential," *Proc. 2006 IEEE/IFIP Netw. Operations Manage. Symp.*, pp.1–4.

[17] Y. Yan, A. El-Atawy, and E. Al-Shaer, "A game-theoretic model for capacity-constrained fair bandwidth allocation," *Int. J. Netw. Manage.*, vol. 18, no. 6, pp. 485–504, Nov. 2008.

[18] M. Charalambides, P. Flegkas, G. Pavlou, *et al.*, "Policy conflict analysis for diffserv quality of service management," *IEEE Trans. Netw. Service Manage.*, vol. 6, no. 1, pp. 15–30, Mar. 2009.

