National Conference on Innovative Trends in Science and Engineering (NC-ITSE'16)
Volume: 4 Issue: 7
ISSN: 2321-8169
229 - 233

Effective Design of High Performance Packet Classification Architecture using HDL

Kranti Hande
ETC Department
P.G. Student of Agnihotri College of Engineering
Wardha, India.
kranti_15sep@rediffmail.com

Prof. Vijay Wadhankar
ETC Department
H.O.D. of Agnihotri College of Engineering
Wardha, India.
vijayhingan@yahoo.com

***Abstract*—** Performing packet classification is challenging since it involves inspection of multiple field against multiple rules with high speed. Packet classification is needed for services such as firewall and services that require capability to distinguish and isolate traffic of packets in different flows for suitable processing. In this work we propose pipelined architecture on FPGA to perform memory efficient, high speed packet classification. In this work, we eliminate use of generation of bit vector algorithm instead of that we are using XNOR gate to generate bit vector which increases memory efficiency. We also use incorporate range search to avoid ruleset expansion occurs by range to prefix conversion. After implementation on Altera Cyclone® IV 4CE115 FPGA device, it is found that this architecture is able to operate at 80 Gbps and supports large rulesets up to 28 K rules using only the on-chip memory with lowest cost and lowest power consumption.

Keywords- p*acket classification;pipeline architecture; ruleset; Altera Quartus II; DE2-115 FPGA Development Board*

*****

I. INTRODUCTION

For many users at once to transmit data quickly data is broken into small chunks called as packets or frames. Packet classification is used to sort packets into different categories by comparing their headers to a list of rules. A flow is used to decide a packet's priority and the manner in which it is processed. Due to the rapid growth of the internet as well as the ruleset size, multi-field packet classification has become challenging because less processing time is available and large amount of memory is required to store data. In existing methods different algorithms are used for packet classification, these methods use range to prefix conversion which require more memory because a single range is converted into prefix format which converts each bit into 0,1 or *(do not care state) causes more number of comparisons with header field, therefore more memory required. In our method we used range matching in which a range of upper and lower bound is provided by the rule and a match is found if the header field match within this range. Therefore it requires only two comparison for each rule i.e.it requires less number of comparison as compare to range to prefix conversion. We are using XNOR gate to eliminate bit vector generation algorithm. Bits of header field and bits of rules are applied to the input of XNOR gate, XNOR gate compare header field and rules and generate resultant bit vector. Due to using XNOR gate and incorporate range search, memory efficiency increases. Previous algorithms are not scalable i.e. there is no same performance for different rules. Now a days ruleset are changing frequently. Our method is ruleset independent therefore there is no problem for changing ruleset. We evaluate pipelined architecture on cyclone IVE FPGA.FPGA allow the use of on chip SRAM which keeps throughput as high as required. Keeping the logic gate and memory on a single chip also has the advantage of allowing for a one-chip solution which further reduces power consumption. FPGA can implement multiple packet processing engines. This further increases throughput. FPGA offers reconfigurability.

II. RELATED WORK

In "A Scalable and Modular Architecture for High-Performance Packet Classification" [1], field is divided into sub-field. Sub field length of k bits is called as a stride. There is independent matching in between bits of the rule with corresponding k bits of an input packet header. Comparison is done in between k bit header field with the lower and upper bounds of the rules of stride. This method is called as incorporate range search. The result of the first stride is passed to the next stride and they are bitwise ANDed to get entire classifier. Highest priority match is extracted by pipelined priority encoder. This algorithm has more than 100 Gbps throughput, moderate memory efficiency and it is independent on ruleset.

In "Ultra-High Throughput Low-Power Packet classification" [2], modified version of the HyperCut packet classification algorithm is used with a new pre-cutting process, high throughput 138.56 Gbps is achieved, low memory requirement, but high power consumption 9.03 W.

In "Scalable Packet Classification on FPGA" [4], decision tree based HyperCut algorithm is used, linear pipeline architecture is used, it is ruleset dependent, throughput is 80 Gbps and more memory is required.

In "Multi-dimensional Packet Classification on FPGA: 100 Gbps and Beyond" [5], Hyper split algorithm is used, it is ruleset dependent, throughput is high, support 50K rules and more memory required.

In "Hardware Based packet Classification for High Speed Internet Routers" [6], TCAM (Ternary Content Addressable memory) is used, it requires more transistor to store same bits than RAM. It has low power efficiency and it is dependent on ruleset. TCAM can match all rules from a ruleset with header field in 1 clock cycle. This is achieved by carrying out parallel comparisons on all stored rules in a single clock cycle. It consume power between 4.86-19.14 watts depending on the TCAM size. Besides the high power consumption, another

drawback for TCAM is its poor storage efficiency of rulesets when using rules containing ranges. This is because a memory word's bits are stored in a 1, 0 or do not care state. This makes TCAM very efficient at storing fields that use longest prefix matching but poor at storing fields that use range matching. In parallel bit vector algorithm each bit of bit vector represents rule of the entire ruleset, each bit of rule compare with bit of header field independently than the other bits in the bit vector. This algorithm has high throughput 115 Gbps and lowest latency 1 clock, but it has low memory efficiency and high power consumption.

In "Field-Split Parallel Architecture Packet Classification" [7], field split algorithm and a novel field-split parallel bit vector (FSBV) architecture for multi-match packet classification is used. This approach requires linear memory which increase with the number of rules. FPGA implementation results showed that the FSBV architecture could store the full set of the current rule and headers using small amount of on-chip resources and it is ruleset dependent, throughput is up to 100 Gbps.

In "Distributed Cross-producing of Field Labels (DCFL)" [11], use independent search engines and operate in parallel manner. Instead of using bit-vectors, DCFL uses a network of efficient aggregation nodes by employing Bloom Filters and encoding intermediate search results. It is highly dependent on ruleset. Its throughput is lowest nearly 19 Gbps.

In "Efficient Packet Classification for Network Intrusion Detection Using FPGA" [12], BV-TCAM algorithm is used, it require more memory, it is ruleset dependent, power consumption is more, throughput is less i.e. 75 Gbps. It is suitable for less number of rules.

In "Algorithms for packet classification" [13], in basic search algorithm, Tuple Space Search algorithm has less throughput, high memory requirement and highly dependent on ruleset. HyperCut packet classification algorithm breaks a ruleset into groups, a decision tree is used to guide a packet based on its header values to the correct group to be searched. This algorithm throughput is very less, it is ruleset dependent.

III. ALGORITHM MODIFICATION

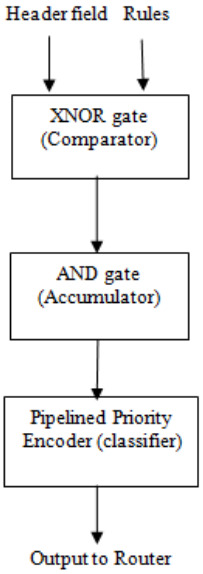

Figure 1. Block diagram

Module A. XNOR Block

Header field bits and rules bits are applied to XNOR gate to get resultant bit vector. XNOR gate compare header bits with rule bits .If header bit is matches with rule bit then resultant bit is set to 1 and if header bit is not matches with rule bit then resultant bit is set to 0.There is no need to split the field bits. Using XNOR we eliminate field split and bit vector algorithm.

Example:

Rule	Header field	XNOR output
1100	1000	1011
1011	1001	1101
1001	0100	0010
1*10	1*10	1111

Module B. AND Block

Resultant bit vectors ANDed together by accumulator. Output of accumulator goes to the priority encoder.

Example:

XNOR output	AND output
1011	0
1101	0
0010	0
1111	1

Module C. Priority Encoder

Output from AND gates are applied to priority encoder, it identifies first bit position which is set to 1 and find out the class. For header field 1010 class 4 is identified.

Example:

Priority Encoder input	Priority Encoder output
0	Class 4
0	
0	
1	

In this modified algorithm we are not using any algorithm to generate bit vector we are using only packet classification algorithm.

IV. MODULAR PIPELINED ARCHITECTURE

We are using incorporate range search method, in this method, two values are loaded at each pipeline stage, per rule the packet header value is checked against both lower and upper bound of rules. These values either can be stored locally in the pipeline stage or stored in stage memory. In the incorporate range search, stages do not require storage of entire bits. In the pipeline architecture each individual bit is independent on the other bits therefore partitioning is possible. Therefore there is no requirement to load all N bits at each pipeline stage, only N/P bits are required to load in pipeline stage, where P is number of partitions. Thus memory bandwidth requirement is decreased. XNOR gate is used for comparison of rules and header field. Rules and header field

 ISSN: 2321-8169
229 - 233

are the inputs of XNOR gate, due to use of XNOR gate we eliminate the generation of bit vectors by field splitting which improves the memory. Bits of header field is used as address to stage memory. The output bit vector of stage memory and the bit-vector generated from the previous stage are bitwise ANDed together and generate final bit vector of the current stage and the priority encoder find out the highest priority match from the resultant bit-vector.

Rules are arranged in order of decreasing priority. Priority encoder identifies first bit positions which is set to one and identify class in one cycle. But when the length of resultant bits increases, time required to find out highest priority match increases. Therefore throughput decreases. To overcome this problem log B N stages pipelined priority encoder is used which operate at high frequency, where N is number of resultant bits and B is number of partitions, the resultant bits are split in a given stage.

V. SOFTWARE AND HARDWARE USED

For designing we are using Altera Quartus II version 9.1 and Modelsim 6.3 software. Modelsim is multi-language HDL simulation environment by Mentor Graphics, for simulation of hardware description languages such as VHDL, Verilog and SystemC includes a built-in C debugger. Modelsim can be used independently, or in conjunction with Altera Quartus or Xilinx ISE. Simulation is performed using the graphical user interface (GUI) or automatically using scripts.

We are using Altera Cyclone IV 4CE115 FPGA DE2-115 board for hardware. It has many features that allow users to implement a wide range of designed circuits, from simple circuits to various multimedia projects. Altera's Cyclone® IV FPGA family provides an ideal platform for high-volume, cost-sensitive applications, helps to increase system bandwidth requirements while lowering system costs. Cyclone IV FPGAs extend the Cyclone FPGA series in providing the market's lowest cost, lowest power FPGAs, with a transceiver variant. Built on an optimized low-power process and utilizing more on-chip hard intellectual property (IP) blocks, Cyclone IV FPGAs consume less total power while lowering system cost.

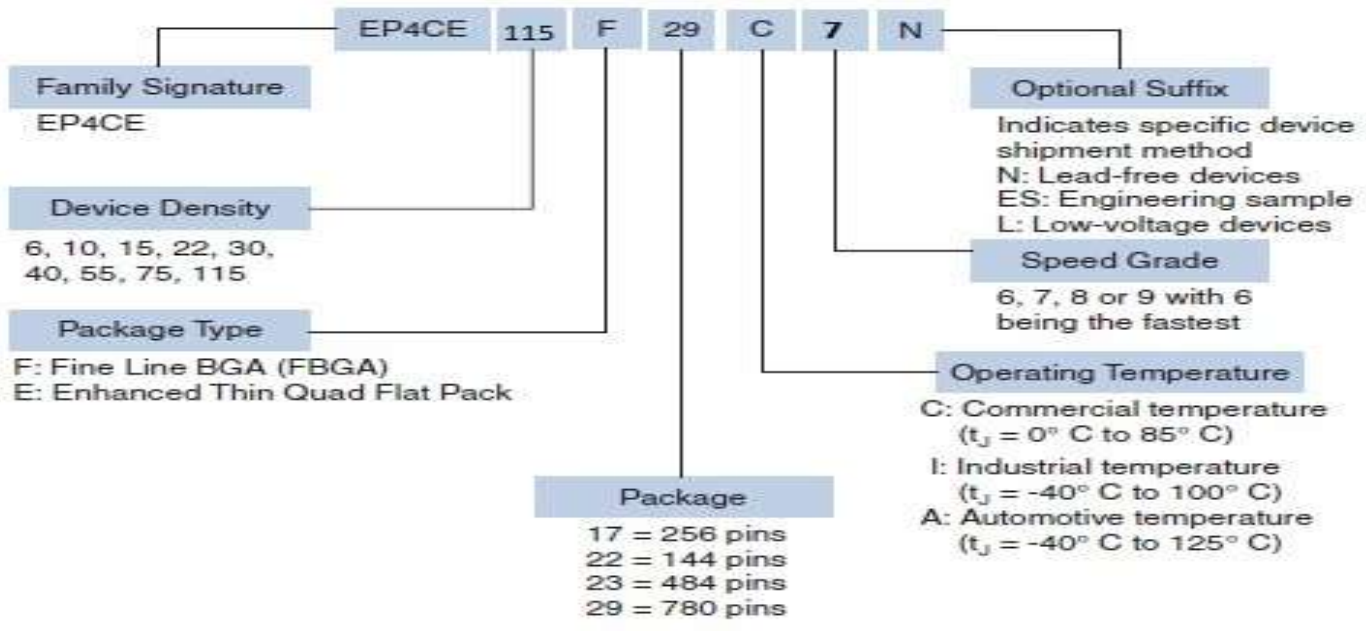

Figure 2. Packaging Ordering Information for the Cyclone IV E Device

The following hardware is provided on the DE2-115 board:

- Altera Cyclone® IV 4CE115 FPGA device
- Blaster (on board) for programming; both JTAG and Active Serial (AS)
- 2MB SRAM
- Two 64MB SDRAM
- 8MB Flash memory
- SD Card socket
- 4 Push-buttons
- 18 Slide switches
- 18 Red user LEDs
- 9 Green user LEDs
- 50MHz oscillator for clock sources
- 24-bit CD-quality audio CODEC with line-in, line-out, and microphone-in jacks
- VGA DAC (8-bit high-speed triple DACs) with VGA-out connector
- TV Decoder (NTSC/PAL/SECAM) and TV-in connector
- 2 Gigabit Ethernet PHY with RJ45 connectors
- USB Host/Slave Controller with USB type A and type B connectors

 ISSN: 2321-8169
229 - 233

- RS-232 transceiver and 9-pin connector
- PS/2 mouse/keyboard connector
- IR Receiver
- 2 SMA connectors for external clock input/output

Cyclone IV EP4CE115F29 device features:

- 432 M9K memory blocks
- 4 PLL
- 3888 Kbits embedded memory
- 114480 logic elements

VI. SIMULATION RESULTS

We designed programs of XNOR, AND gate, priority encoder, memory, one rule classifier, package classification in VHDL. Then we combined all these programs to form complete system program. We compile and simulate complete system program with the help of test benches on Modelsim 6.3.The following are the classifier input and output.

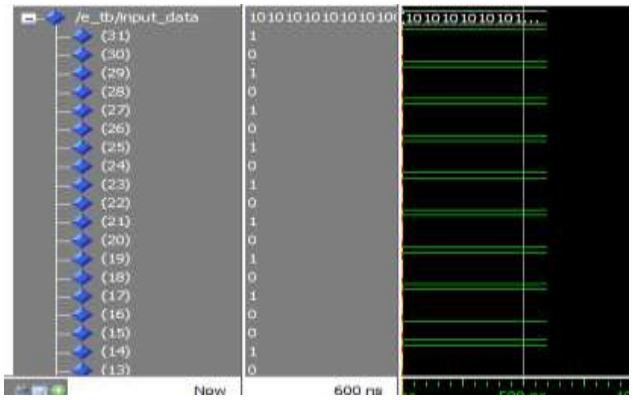

Figure.3 Classifier input

32 bits of header field is applied to classifier program as input to find the class. After simulation it is found that class 2 is occurs maximum times, therefore for 32 bit header field class 2 is the output in 7 bits.

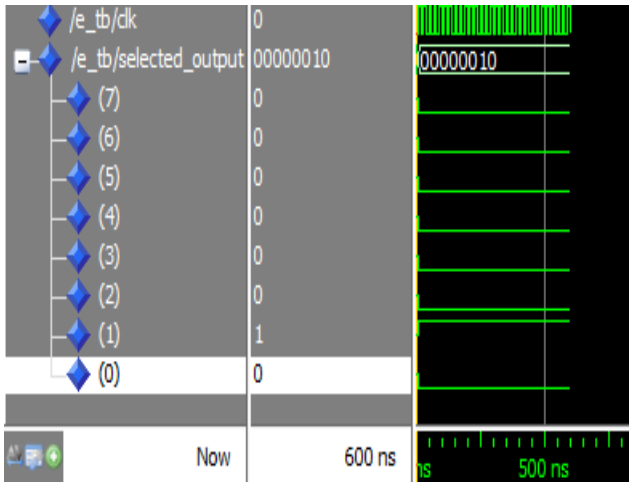

Figure 4. Classifier output

VII. IMPLEMENTATION

1. Open Quartus II software, by double clicking on Quartus II icon.
2. Take DE2-115 Development Board.
3. Connect 12V supply through adapter to turn ON/OFF the board.
4. Connect USB cable to laptop/computer and also to the USB Blaster of the Board.
5. Add program file to project.
6. Select device family name Cyclone® IV 4CE115 FPGA.
7. Analysis the program.
8. Assign a pins at pin planner, by using user manual of DE2-115 board
9. Generate programming files by assembler.
10. On the FPGA board.
11. Select USB blaster.
12. You can use JTAG method of programming, In this method of programming, named after the IEEE standards Joint Test Action Group, the configuration bit stream is downloaded directly into the Cyclone IV E FPGA. The FPGA will retain this configuration as long as power is applied to the board; the configuration information will be lost when the power is turned off.
13. Click start.
14. After successful downloading change the inputs i.e. header field bits with the help of knobs.
15. Observe the class of packet classification with the help of glowing LED output.

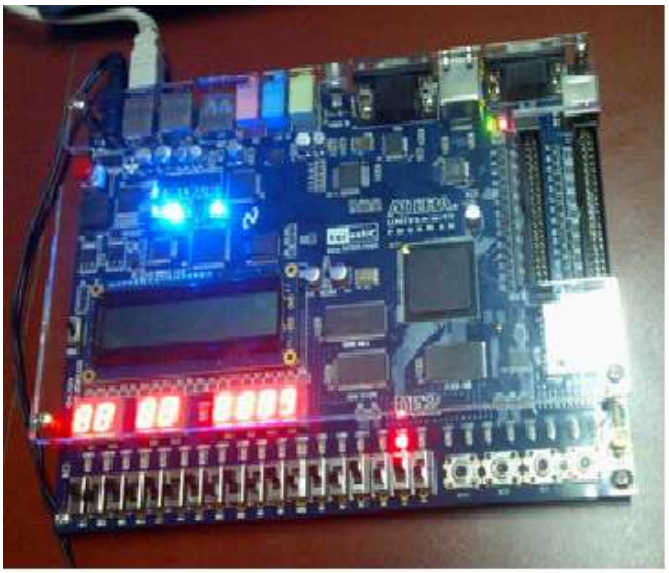

Figure5. Output on FPGA

After implementation on FPGA it is found that after application of same input which is applied to classifier program on Modelsim 6.3, output is class 2 indicated by glowing LED on FPGA kit.

VIII. PERFORMANCE RESULTS

We evaluate the performance of classifier on Altera Cyclone IV 4CE115 FPGADE2-115 board. The classifier used a modified version of the algorithm using XNOR gate that has been modified so that it is better suited to hardware implementation. The performance of hardware is measured in the

number of packets forwarded per second (PPS) or the number of data bits forwarded per second (bps).With the Cyclone IV E 115 Altera classifier can achieve a maximum clock speed of 250 MHz with 4 phase locked loop and global clock (GCLK) network. This is possible because each of its engines can classify a packet in two memory accesses and dual port memory is used, allowing two memory accesses to be made per clock cycle. A maximum throughput for classifier having minimum packet size which is 40 bytes is 80 Gbps.

With the help of another version of Altera or by using Xilinx FPGA we can achieve higher throughput but this version has better throughput according to the cost and power consumption. Static power is almost constant while dynamic power increases with the increasing partition size because more clock has to be given to more number of units. The power consumption is measured by simulations, with the software Quartus 2 Power Play Power Analyzer. Power consumption increases with increase in number of ruleset. Its all applications are under 1.5 W total power.

In our method, we used range matching in which a range of upper and lower bound is provided by the rule and a match is found if the header field match within this range. Therefore it requires only two comparison for each rule i.e.it requires less number of comparison as compare to range to prefix conversion. We are using XNOR gate to eliminate bit vector generation algorithm which required more memory, therefore this algorithm requires less amount of memory as compare to any algorithm. Altera Cyclone IV 4CE115 FPGA has 432 M9K on chip memory blocks. Each block is 256*36 RAM block and contains 9,216 programmable bits. It supports large rulesets up to 28 K rules using only the on-chip memory. Our method is ruleset independent.

IX. CONCLUSION AND FUTURE WORK

We used XNOR gate to generate bit vector which increases memory efficiency. We also used incorporate range search to avoid ruleset expansion occurs by range to prefix conversion. Our method is ruleset independent. Implementation of our algorithm on the Cyclone IVE also achieves high throughput, with its memory efficiency, obtaining a maximum clock speed of 250 MHz using 4 phase locked loop. PLL and global clock (GCLK) network can dynamically reconfigure Cyclone IV device in user mode to change the clock frequency or phase. This allows to achieve throughput 80 Gbps or 250 Mpps with memory efficiency at lowest power. In the future, we plan to measure performance of BV-XNOR on latest version of FPGA and modify the packet classification algorithm to improve efficiency.

ACKNOWLEDGEMENT

I would like to thanks Dr.C.M.Jadhao, Principal, MGI-COET, Shegaon and Prof.S.S.Mhaske, HOD (ENTC), MGI-COET, Shegaon, for their support.

REFERENCES

[1] Thilan Ganegedara, Weirong Jiang, and Viktor K. Praasanna "A Scalable and Modular Architecture for High-Performance Packet Classification," IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 25, NO. 5, MAY 2014.

[2] Kennedy and Xiaojun Wang, "Ultra-High Throughput Low Power Packet Classification" IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, VOL. 22, NO. 2, FEBRUARY 2014

[3] A. Sanny, T. Ganegedara, and V. Prasanna, ''A Comparison of Ruleset Feature Independent Packet Classification Engines on FPGA,'' in Proc. IEEE IPDPS RAW, 2013, pp. 124-133.

[4] Weirong Jiang, and Viktor K. Prasanna, "Scalable Packet classification on FPGA" in Springer-Verlag, 2011.

[5] Yaxuan Qi, Jeffrey Fong, Weirong Jiang, Bo Xu , Jun Li, Viktor prasanna,"Multi-dimensional packet classification on FPGA: 100 Gbps and beyond" IEEE INFOCAM, 2010

[6] C.R. Meiners, A.X. Liu, and E. Torng, "Hardware Based packet Classification for High Speed Internet Routers" Berlin, Germany, Springer Verlag, 2010.

[7] W. Jiang and V.K. Prasanna, ''Field-Split Parallel Architecture Packet Classification,'' Annu. SPAA, 2009. 188-196.

[8] M. Faezipour and M. Nourani, ''Wire-Speed TCAM-Based Architectures for Multimatch Packet Classification,'' IEEE Trans.Comput., vol. 58, no. 1, pp. 5-17, Jan. 2009.

[9] Heeyeol Yu and Rabi Mahapatra,"A Power- and Throughput-Efficient Packet Classifier with n Bloom Filters" IEEE TRANSACTION ON COMPUTERS .2008.

[10] G. Jedhe, A. Ramamoorthy, and K. Varghese, ''A Scalable High Throughput Firewall in FPGA," Apr. 2008, pp. 43-52.

[11] D. Taylor and J. Turner, ''Scalable Packet Classification Using Distributed Cross Producing of Field Labels,'' in Proc. 24th Annu.Joint IEEE INFOCOM, Mar. 2005, vol. 1, pp. 269-280.

[12] H. Song and J.W. Lockwood, ''Efficient Packet Classification Network Intrusion Detection Using FPGA,'' in Proc. ACM/SIGDA 13th Int'l Symp. FPGA, 2005, pp. 238-245.

[13] P.Gupta and N. McKeown, "Algorithms for packet classification,"IEEE Network, vol. 15, no. 2, pp. 24–32, 2001.

[14] P. Gupta and N. McKeon, ''Classifying Packets with Hierarchical Intelligent Cuttings,'' IEEE Micro Lookups with IncrementalUpdates,'' SIGCOMM Jan. /Feb. 2000.

[15] Pankaj Gupta and Nick McKeown "Packet Classification for multiple field" Proc. Sigcomm, Computer Communication Review, vol. 29, no. 4, pp 147-60, September 1999, Harvard University.

[16] T.V. Lakshman and D. Stiliadis, ''High-Speed Policy-Based Packet Forwarding Using Efficient Multi-Dimensional Range Matching,'' SIGCOMM Comput. Commun. Rev., vol. 28, no. 4, pp. 203-214, Oct. 1998.