Buletinul Ştiinţific al Universităţii "Politehnica" din Timişoara

Seria ELECTRONICĂ şi TELECOMUNICAŢII
TRANSACTIONS on ELECTRONICS and COMMUNICATIONS

Tom 50(64), Fascicola 2, 2005

Modern Network Algorithms

Miranda Nafornita[1], Florin Chis[2]

Abstract – In the future, the routing will be based not only on destination host numbers but also on source host or even source users, as well as destination URLs (Universal Resource Locators) and specific business policies. Modern routers and switches will have to use class of services and QoS to determine the path to particular applications for particular end users. All this requires the use of layers 4, 5 and above.
Keywords: routing, prefix lookup, packet classification

I. INTRODUCTION

Most users deal with domain names, but these names are translated to an IP address by a directory service called DNS before packets are sent. Internet prefixes are defined using bits and not alphanumerical characters, of up to 32 bits in length. Internet began with a simple hierarchy in which 32-bit addresses were divided into a network address and a host number; routers only stored entries for the networks. For flexible address allocation, the network address came in variable sizes: Class A (8 bits), Class B (16 bits), and Class C (24 bits).

IP *prefixes* are often written in *dot-decimal notation*; so the notation is equivalent to a binary string. To use more efficiently address space, Internet prefixes are variable length so the second way to denote a prefix is by *slash notation* A/L (A denotes a 32 IP address in dot-decimal notation and L denotes the length of prefixes). Shorter prefixes can be assigned to areas with a large number of endpoints and larger prefixes to those with a few endpoints. A third common way to describe prefixes is to *use a mask* in place of an explicit prefix length. The last two ways are more compact for writing down large prefixes [10].

To cope with exhaustion of Class B addresses, the Classless Internet Domain Routing (CIDR) scheme assigns for new organizations multiple contiguous Class C addresses that can be aggregated by a common prefix. This reduces core router table size. The potential depletion of address space has led Internet registries to be very conservative in the assignment of IP addresses. Many organizations are coping with these sparse assignments by sharing a few IP addresses among multiple computers, using schemes such a network address translation, or NAT.

Thus CIDR and NAT have helped the Internet handle exponential growth with a finite 32-bit address space. While there are plans for a new IP (IPv6) with a 128-bit address, the effectiveness of NAT in the short run and the complexity of rolling out a new protocol have slowed down the IPv6 deployment. Despite this, a new world containing billions of wireless sensors may lead to an IPv6 resurgence.

II. PREFIX LOOKUP

The decision to deploy CIDR helped save the Internet, but it has introduced the complexity of *longest-matching-prefix lookup*. Three kinds of matches are allowed: exact match, prefix match and range match. *Exact-match lookups*: the header field of the packet should exactly match the field and they represent the simplest form of database query. A query specifies a key K and the response returns the state information associated to this key. Exact-match queries are easily implemented using well-known techniques, such as binary search and hash tables. But in the networking context, the models and metrics for lookups are different. The lookups must be completed in the time to receive a packet, the use of memory references rather than processing as measure of speed, and the potential use of hardware speedups. Exact-match lookups are crucial for a very important networking function called *bridging*, often integrated within routers [8].

Prefix–match: the field should be a prefix of the header field. A packet arriving on an input link of a router carries a 32-bit Internet (IP) address. The processor consults a forwarding table to determine the output link for the packet, which contains a set of *prefixes* with their corresponding output link. The packet is matched to the longest prefix that matches the destination address in the packet, and the packet is forwarded to the corresponding output link. The task of determining the output link is called *address lookup*, and must be implemented at very high speeds (until gigabit or terabit per second).

In a *range match*, the header values should lie in the specified range [8].

[1] Politehnica University of Timisoara, Communications dept.
Bd. V. Parvan no. 2, 300223 Timisoara, e-mail miranda.nafornita@etc.upt.ro
[2] Politehnica University of Timisoara, Communications dept.

Traffic distributions, memory trends and database sizes motivate the requirements for lookup schemes. A study of backbone traffic shows over 250.000 concurrent flows of short duration and this number is increasing, that means caching solutions do not work well. Roughly half the packets received by a router are minimum-size TCP acknowledgements. Hence, the router must be able to perform prefix lookups in the time to forward a minimum-size packet. Assuming wire speed forwarding, forwarding a 40-byte packet should take no more than 320 nsec at 1 Gbps, 32 nsec at 10 Gbps, and 8 nsec at 40 Gbps.

Clearly, the most crucial metric for a lookup scheme is lookup speed. Cost of computation is dominated by memory access time, so the simplest measure of lookup speed is the worst-case number of memory access. The possible use of host routes (full 32-bit addresses) and multicast routes means that the backbone routers will have prefix databases till 1 million prefixes. Unstable routing-protocol implementation can lead to requirements for updates (add or delete a prefix) on the order of milliseconds, a several orders of magnitude below the lookup requirement, allowing to pre-compute information to speed up lookup, at the cost of longer update times. Memories can be cheap and slow (DRAM access time： 60 nsec) or faster and expensive (SRAM off / on-chip memory, 1-10 nsec). So, the interesting *metrics for lookup* schemes are lookup speed, memory, and update time [5].

The longest matching prefix is a very complex problem. In virtual circuit networks, like ATM [13], when a source wishes to send data to a destination, a call is set up. The virtual circuit identifier VCI at each router is a moderate-size integer that is easy to lookup but the cost is a round-trip delay for call setup. The ATM has a previous hop switch pass an index into next hop switch, pre-computed just before data is send by the previous hop switch. The same idea was used in datagram networks such as Internet *to finesse the need for prefix lookup*: *tag switching* and *flow switching.*

Tag switching is a first proposal to finesse lookup. In threaded indices each router passes an index into the next router's forwarding table, thereby avoiding prefix lookup. The indexes are pre-computed by the routing protocol whenever the topology changes. The main differences between threaded indices and virtual circuit indices are: 1) threaded indices are per destination and not per active source-destination pair as in virtual circuit networks such as ATM; 2) threaded indexes are pre-computed by the routing protocol whenever the topology changes. Cisco later introduced tag switching, which is similar in concept to threaded indices, except tag switching also allows router to pass a stack of tags (indices) for multiple routers downstream. Both schemes do not deal well with hierarchies. The last backbone router has only one aggregated routing entry for the entire destination domain and can thus pass only one index for all subnets in that domain. The adopted solution was to require ordinary IP lookup at domain entry points. Tag switching, developed today in a more general form, is *multiprotocol label switching*, MPLS. Neither tag switching nor MPLS completely avoid the need for ordinary IP lookups and each adds a complexity.

Flow switching is a second proposal to finesse lookup and also relies on a previous hop router to pass an index into the next hop router. Unlike tag switching, these indexes are computed on demand when data arrives, and then are cached. Eventually, IP forwarding can be completely avoided in the switched portion of a sequence of flow-switched routers. Flow-switching seems likely to work poorly in the backbone, because backbone flows are short lived and exhibit poor locality. On the other hand the current use of circuit-switched optical switches to link core routers, the underutilization of backbone links and increase of optical bandwidth make possible the resurrection of flow switching based on TCP connections.

The most important lookup algorithms are as follows. If memory is not an issue, the fastest scheme is one called recursive flow classification (RFC). If memory is an issue, a simple scheme that works well for classifiers up to around 5000 rules is the Lucent bit vector scheme. For larger classifiers, the best trade-off between speed and memory is provided by decision tree scheme. Unfortunately, all these algorithms are based on heuristics and cannot guarantee performance on all databases. If guaranteed performance is required for more than two field classifiers, there is no alternative but to consider hardware schemes such as ternary CAM (content addressable memory).

III. PACKET CLASSIFICATION

Future routers and switches will have to use class of services and QoS to determine the path to particular applications for particular end users. All this requires the use of layers 4, 5 and above. This new vision of forwarding is called *packet classification* or *layer 4 switching*, because routing decision can be based on headers available at layer 4 or higher in the OSI architecture. Other fields a router may need to examine include source address (to forbid or provide different service to some source networks), port fields (to discriminate between traffic types), and even TCP flags (to distinguish between externally and internally initiated connections). Besides security and QoS, other functions that require classification include network address translation (NAT), metering, traffic shaping, policing, and monitoring.

The packet classification problem is to *determine the lowest-cost matching rule* for each incoming message at a router. The information relevant to a lookup is contained in K distinct header fields in each message. The *classifier*, or *rule database*, consists of a finite set of N rules, each rule being a combination

of K values, one of each header field. Each field in a rule is allowed the three kinds of matches. Exact match, where the header field of the packet should exactly match the rule field, is useful for protocol and flag fields. Prefix match, where the rule field should be a prefix of the header field could be useful for blocking access from a certain subnetwork. Range match, where the header values should lie in the range specified by the rule, can be useful for specifying port number range.

Each rule has an associated directive, which specifies *how to forward* the packet matching this rule. The directive specifies if the packet should be blocked or forwarded. If the packet should be forwarded, the directive specifies the outgoing link to which the packet is sent, and, perhaps, also a queue within that link if the message belongs to a flow with bandwidth guarantees. A packet is said to *match a rule*, if each field of a packet matches the corresponding field of the rule. The match type is implicit in the specification of the field. Since a packet may match multiple rules in the database, each rule in the database is associated with a nonnegative number, *the cost*. Ambiguity is avoided by returning the least-cost rule matching the packet's header. The cost function generalizes the implicit precedence rules that are used in practice to choose between multiple matching rules.

Several variants of packet classification have already established on Internet. First, many routers implement *firewalls* at trust boundaries, such as the entry and exit points of a corporate network. A firewall database consists of a series of packet rules that implement security policies. Rules are placed in the database in a specific linear order, where each rule takes precedence over a subsequent rule. Thus, the goal there is to find the *first* matching rule. The same effect can be achieved by making cost equal to the position of rule in database. A general firewall could arbitrarily interleave rules that allow packets with rules that drop packets. Second, the need for predictable and guaranteed service has led to proposals for *reservation protocols*, such as DiffServ, that reserve bandwidth between source and destination. Third, routing based on traffic type has become more stringent recently.

Packet classification unifies the forwarding function required by firewalls, resource reservation, QoS routing, unicast routing and multicast routing. In classification, the forwarding database of a router consists of a potentially large number of rules on key header fields. A given packet header can match multiple rules. Each rule has a given cost and the packets are forwarded using the least-cost matching rule.

IV. MPLS - MULTIPROTOCOL LABEL SWITCHING

Originally introduced to speed up lookups, MPLS is now a mechanism for providing flow differentiation quality of service (QoS). Another major feature of MPLS is the ability to place IP traffic on a defined path through the network providing bandwidth guarantees and differentiated service (DiffServ) features for specific user application or flow [8,10, 11].

Since a VCI provides a simple label to quickly distinguish a flow, a label allows a router to easily isolate a flow for special service. MPLS uses labels to finesse the need for packet classification, a much more difficult problem than prefix lookup. MPLS is used for the core router today, although prefix matching is still required. IP networks with connectionless operation, for traffic engineering purposes, becomes more connection-oriented network, where the path between the source and the destination is pre-calculated based on user specifics. To speed up the forwarding schemes, an MPLS device uses labels rather than address matching to determine the next hop for a received packet. To provide traffic engineering, tables are used, tables that represent the levels of QoS the network can support. The tables and the labels are used together to establish an end-to-end path called a label switched path (LSP). Traditional IP routing protocols (e.g.: OSPF-open shortest path first, IS-IS intermediate system to intermediate system), and extension to existing signaling protocols (RSVP-resource reservation protocol and CR-LDP-constraint-based routing label distribution protocol) comprise the suite of MPLS protocols.

MPLS is based on the following ideas: a) Forwarding information (label) separate from the content of IP header; b) A single forwarding paradigm (label swapping), multiple routing paradigms; c) multiple link-specific realizations of the label swapping forwarding paradigm "shim", virtual connection/path identifier (VCI/VPI), frequency slot (wavelength), time-slot; d) The flexibility to form forwarding equivalence classes (FECs); e) A forwarding hierarchy via label stacking [14].

The separation of forwarding information from the content of the IP header allows MPLS to be used with devices such as OXCs (optical cross-connect), whose data plane cannot recognize the IP header. LSRs (label switch routers) forward data using the label carried by the data. This label, combined with the port of the switch where data was received, is used to determine the output port and outgoing label for the data. The MPLS control plane operates in terms of label swapping and forwarding paradigm abstraction. At the same time, the MPLS control plane allows multiple link-specific realizations of this abstraction. For example [15], a wavelength could be viewed as an implicit label. Finally, the concept of a forwarding hierarchy via label stacking enables interaction with devices that can support only a small label space. This property of MPLS is essential in the context of OXCs and DWDM (Dense Wavelength Division

Multiplexing) since the number of wavelengths (which acts as labels) is not very large.

MPLS fast path forwarding is as follows. A packet with an MPLS header is identified, a 20-bit label is extracted, and the label is looked up in a table that maps the label to a forwarding rules. The forwarding rule specifies a next hop and also specifies the operations to be performed on the current set of labels in the MPLS packet. These operations can include removing or adding labels. Router MPLS implementations have to impose some limits to guarantee wire speed forwarding. The label space may be dense, supporting a smaller number of labels than 2^{20} (this allows a smaller amount of lookup memory, avoiding a hash table) and limiting the number of label-stacking operations that can be performed on a single packet.

The MPLS framework includes significant applications such a *constraint-based routing*. Constraint-based routing is a combination of extensions to existing IP link-state routing (e.g., OSPF and IS-IS) with RSVP or CR-LDP like the MPLS control plane, and the Constrained Shortest-Path-First (CSPF) heuristic. The extensions to OSPF and IS-IS allow nodes to exchange information about network topology, resource availability and even policy information. This information is used by the CSPF heuristic to compute paths subject to specified resource and/or policy constraints. For example, either RSVP-TE (TE – Traffic Engineering) or CR-LDP is used to establish the label forwarding state along the routes computed by CSPF-based algorithm; this creates the LSP. The MPLS data plane is used to forward the data along the established LSPs. Constraint-based routing is used today for two main purposes: traffic engineering and fast reroute. With suitable network design, the constraint-based routing of IP/MPLS can replace ATM as the mechanism for traffic engineering. Fast reroute offers an alternative to SONET as a mechanism for protection/restoration.

V. GENERALIZED MPLS (GMPLS)

GMPLS extends MPLS to provide the control plane (signaling and routing) for devices that switch in any of these domains: packet, time, wavelength and fiber [7]. This common control plane promises to simplify network operation and management by automating end-to-end provisioning of connections, managing network resources and providing the level of QoS that is expected in the new applications. GMPLS is promoted as a major technology for the automation of network operations [3].

Because of the large installed base of SONET/SDH network infrastructure and circuit-oriented TDM-based networks there is a need for efficient interworking between traditional circuit-oriented networks and IP/MPLS networks. The most promising technology able to meet these requirements is the GMPLS protocol suite. GMPLS extends the label switching introduced in MPLS. The latter, originally introduced in IP networks to improve forwarding performance by eliminating time-consuming longest prefix matching, has shifted toward improved resiliency, enhanced QoS, and traffic engineering capabilities [1], [2], [3], [12]. GMPLS has been specifically designed to extend capabilities offered by MPLS network elements that have non-packet-based forwarding engines (from packet and frame/cell switching technologies to circuit switching technologies, including SONET/SDH). It encompasses the entire range from packet-switching-capable devices up to fiber-switching-capable devices. The GMPLS architecture specifies all the protocol capabilities in term of signaling (RSVP-TE Resource-Reservation-Protocol-Traffic-Engineering), routing (OSPF-TE), and link-management (LMP Link-Management-Protocol) [4].

The main idea of GMPLS is to extend the original MPLS scheme that needs to recognize the packet boundaries and extract the label information before switching the packet, to a scheme that can perform switching without depending on recognizing the packet boundaries or the header information. Such a scheme must depend on some other optical properties (label mapping space) to find out the forwarding class for a packet before switching it to destination. The application of GMPLS can affect the future of IP-over-WDM networks resulting in reducing the number of layers, which can also reduce the cost, complexity and processing overhead in optical backbones. The current standards for GMPLS define five label mapping spaces, namely: Packet Switch Capable (PSC), Layer 2 Switch Capable (L2SC), Time-slot Switch Capable (TSC), Wavelength Switch Capable (LSC), and Fiber Switch Capable (FSC). Only the last two layers (LSC and FSC) can be utilized in all-optical switching device. The remaining label mapping spaces require optical to electrical conversion, which limits the device speed to a level much less than achievable through an all-optical switching device.

IP-based backbone networks are gradually moving towards a network model consisting of high-speed routers that are flexibly interconnected by lightpaths set up by an optical transport network consisting of WDM links and optical cross-connects. Recovery mechanisms at both network layers will be crucial to reach the high availability requirements of critical services. GMPLS protocol suite can provide a distributed control plane that can be used to deliver rapid and dynamic circuit provisioning of end-to-end optical lightpaths.

VI. REMARKS

In recent years the main focus of transport network evolution has been on increasing transport capacities and introducing data networking technologies and interfaces (e.g. Gigabit Ethernet). This evolution is complemented by outgoing initiatives to reduce the operational effort and

accordingly the costs of network operations. GMPLS together with standardized interfaces like user-network and network-network interfaces (UNI/NNI) are automating the operation of telecom networks. They allow services to be provided efficiently and improve the resilience of networks. For service provisioning (or switched connections), the new approach is that the connections are being set up by the client without operator interaction. This speeds up the provisioning process and reduces effort for the network operator.

REFERENCES

[1] J.T. Park, "Resilience in GMPLS Path management: Model and Mechanism", IEEE Comm. Magazine, July, 2004, pp. 128-135.

[2] D. Papadimitriou, D.Verchere, "GMPLS-Network Interface in Support of End-to-End Rerouting", IEEE Comm. Magazine, July, 2005, pp. 35-43.

[3] A. Banerjee, J. Drake, J.P. Lang, B. Turner, K. Kompelia, Y. Rekhter, "Generalized Multiprotocol Label Switching: An Overview of Routing and Management Enhancements", IEEE Comm. Magazine, Jan, 2001, pp. 144-150.

[4] S. Pasqualini, A. Kirstader, A. Iselt, R. Chahine, S.Vervrugge, D.Colle, M.Pickavet, P.Demeester, "Influence of GMPLS Provider Operational Expenditures: A Quantitative Study", IEEE Comm. Magazine, July, 2005, pp. 28-34.

[5] A. Bianco, M. Atiquzzaman, G.S. Kuo, "Next Generation Switching and Routing", IEEE Comm. Magazine, Jan 2005, pp. 86-87.

[6] Bart Puype et al., "Benefits of GMPLS for Multilayer Recovery", IEEE Comm. Magazine, July, 2005, pp. 51-59.

[7] T. Khattab, H. Alnuweiri, "Optical GMPLS Networks with Code Switch Capable Layer for Sub-Wavelength Switching", IEEE 2004, {tkhattab, hussein}@ece.ubc.ca.

[8] "Generalized Multiprotocol Label Switching, GMPLS", http://www.iec.org.

[9] G. Varghese, "Network Algorithmics", Morgan Kaufman Publishers, 2005.

[10] T.Li, "MPLS an Evolving Internet Architecture", IEEE Comm. Magazine, vol.14, no 2, March/April, 2002.

[11] T.M.Chen, "Reliable Services in MPLS", IEEE Comm. Magazine, vol.37, no 12, Dec, 2002.

[12] X. Xiao, A. Hannan, et al., "Traffic Engineering with MPLS in the Internet", IEEE Comm. Magazine, vol.14, no 2, March/April, 2002.

[13] M. Bocci, J. Guillet, "Operation, Administration and Maintenance in MPLS Networks", IEEE Comm.Magazine, vol.42, no 10, Oct, 2004.

[14] D. Cavendish, H. Ohta, H. Rakotorano, "Framework for MPLS-based Control of Optical SDH/SONET Networks", IEEE Comm. Magazine, vol.14, no 2, March/April,2002.

[15] G. Bernstein, E. Mannie, V. Sharma, "MPLS an Evolving Internet Architecture", IEEE Comm. Magazine, vol.15, no 4, July/Aug, 2001.