Tag Switching

Background

Rapid changes in the type (and quantity) of traffic handled by the Internet and the explosion in the number of Internet users is putting an unprecedented strain on the Internet's infrastructure. This pressure is mandating new traffic-management solutions. *Tag switching* is aimed at resolving many of the challenges facing an evolving Internet and high-speed data communications in general. This chapter summarizes basic tag-switching operation, architecture, and implementation environments. Tag switching currently is addressed by a series of Internet drafts, which include:

- "Tag Distribution Protocol," P. Doolan, B. Davie, D. Katz
- "Tag Switching Architecture Overview," Y. Rekhter, B. Davie, D. Katz
- "Use of Flow Label for Tag Switching," F. Baker, Y. Rekhter
- "Use of Tag Switching With ATM," B. Davie, P. Doolan, J. Lawrence, K. McCloghrie
- "Tag Switching: Tag Stack Encoding," E. Rosen, D. Tappan, D. Farinacci

Tag-Switching Architecture

Tag switching relies on two principal components: forwarding and control. The forwarding component uses the tag information (tags) carried by packets and the tag-forwarding information maintained by a tag switch to perform packet forwarding. The control component is responsible for maintaining correct tag-forwarding information among a group of interconnected tag switches. Details about tag switching's forwarding and control mechanisms follow.

Forwarding Component

The forwarding paradigm employed by tag switching is based on the notion of *label swapping*. When a packet with a tag is received by a tag switch, the switch uses the tag as an index in its Tag Information Base (TIB). Each entry in the TIB consists of an incoming tag and one or more subentries (of the form outgoing tag, outgoing interface, outgoing link-level information). If the switch finds an entry with the incoming tag equal to the tag carried in the packet, then, for each component in the entry, the switch replaces the tag in the packet with the outgoing tag, replaces the link-level information (such as the MAC address) in the packet with the outgoing link-level information, and forwards the packet over the outgoing interface.

From the above description of the forwarding component, we can make several observations. First, the forwarding decision is based on the exact-match algorithm using a fixed-length, fairly short tag as an index. This enables a simplified forwarding procedure, relative to longest-match forwarding traditionally used at the network layer.

This, in turn, enables higher forwarding performance (higher packets per second). The forwarding procedure is simple enough to allow a straightforward hardware implementation. A second observation is that the forwarding decision is independent of the tag's forwarding granularity. The same forwarding algorithm, for example, applies to both unicast and multicast: a unicast entry would have a single (outgoing tag, outgoing interface, outgoing link-level information) subentry, while a multicast entry might have one or more subentries. This illustrates how the same forwarding paradigm can be used in tag switching to support different routing functions.

The simple forwarding procedure is thus essentially decoupled from the control component of tag switching. New routing (control) functions can readily be deployed without disturbing the forwarding paradigm. This means that it is not necessary to re-optimize forwarding performance (by modifying either hardware or software) as new routing functionality is added.

Tag Encapsulation

Tag information can be carried in a packet in a variety of ways:

- As a small "shim" tag header inserted between the Layer 2 and the network-layer headers
- As part of the Layer 2 header, if the Layer 2 header provides adequate semantics (such as ATM)
- As part of the network-layer header (such as using the Flow Label field in IPv6 with appropriately modified semantics)

As a result, tag switching can be implemented over any media type, including point-to-point links, multi-access links, and ATM. The tag-forwarding component is independent of the network-layer protocol. Use of control component(s) specific to a particular network-layer protocol enables the use of tag switching with different network-layer protocols.

Control Component

Essential to tag switching is the notion of binding between a tag and network-layer routing (routes). Tag switching supports a wide range of forwarding granularities to provide good scaling characteristics while also accommodating diverse routing functionality. At one extreme, a tag could be associated (bound) to a group of routes (more specifically to the network-layer reachability information of the routes in the group). At the other extreme, a tag could be bound to an individual application flow (such as an RSVP flow), or it could be bound to a multicast tree.

The control component is responsible for creating tag bindings and then distributing the tag-binding information among tag switches.

The control component is organized as a collection of modules, each designed to support a particular routing function. To support new routing functions, new modules can be added. The following describes some of the modules.

Destination-Based Routing

With destination-based routing, a router makes a forwarding decision based on the destination address carried in a packet and the information stored in the Forwarding Information Base (FIB) maintained by the router. A router constructs its FIB by using the information the router receives from routing protocols, such as OSPF and BGP.

To support destination-based routing with tag switching, a tag switch participates in routing protocols and constructs its FIB by using the information it receives from these protocols. In this way, it operates much like a router.

Three permitted methods accommodate tag allocation and Tag Information Base (TIB) management:

- Downstream tag allocation
- Downstream tag allocation on demand
- Upstream tag allocation

In all cases, a tag switch allocates tags and binds them to address prefixes in its FIB. In downstream allocation, the tag that is carried in a packet is generated and bound to a prefix by the switch at the downstream end of the link (with respect to the direction of data flow). In upstream allocation, tags are allocated and bound at the upstream end of the link. On-demand allocation means that tags are allocated and distributed by the downstream switch only when requested to do so by the upstream switch.

Downstream tag allocation on demand and upstream tag allocation are most useful in ATM networks. In downstream allocation, a switch is responsible for creating tag bindings that apply to incoming data packets, and receiving tag bindings for outgoing packets from its neighbors. In upstream allocation, a switch is responsible for creating tag bindings for outgoing tags, such as tags that are applied to data packets leaving the switch, and receiving bindings for incoming tags from its neighbors. Operational differences are outlined in the following summaries.

Downstream Tag Allocation

With downstream tag allocation, the switch allocates a tag for each route in its FIB, creates an entry in its TIB with the incoming tag set to the allocated tag, and then advertises the binding between the (incoming) tag and the route to other adjacent tag switches. The advertisement can be accomplished by either piggybacking the binding on top of the existing routing protocols, or by using a separate Tag-Distribution Protocol (TDP). When a tag switch receives tag-binding information for a route, and that information was originated by the next hop for that route, the switch places the tag (carried as part of the binding information) into the outgoing tag of the TIB entry associated with the route. This creates the binding between the outgoing tag and the route.

Downstream Tag Allocation on Demand

For each route in its FIB, the switch identifies the next hop for that route. It then issues a request (via TDP) to the next hop for a tag binding for that route. When the next hop receives the request, it allocates a tag, creates an entry in its TIB with the incoming tag set to the allocated tag, and then returns the binding between the (incoming) tag and the route to the switch that sent the original request. When the switch receives the binding information, the switch creates an entry in its TIB, and sets the outgoing tag in the entry to the value received from the next hop.

Upstream Tag Allocation

With upstream tag allocation, a tag switch allocates a tag for each route in its FIB whose next hop is reachable via one of these interfaces (if it has one or more point-to-point interfaces), creates an entry in its TIB with the outgoing tag set to the allocated tag, and then advertises to the next hop (via TDP) the binding between the (outgoing) tag and the route. When a tag switch that is the next hop receives the tag-binding information, the switch places the tag (carried as part of the binding information) into the incoming tag of the TIB entry associated with the route.

After a TIB entry is populated with both incoming and outgoing tags, the tag switch can forward packets for routes bound to the tags by using the tag-switching forwarding algorithm.

When a tag switch creates a binding between an outgoing tag and a route, the switch, in addition to populating its TIB, also updates its FIB with the binding information. This enables the switch to add tags to previously untagged packets. Observe that the total number of tags that a tag switch must maintain can be no greater than the number of routes in the switch's FIB. Moreover, in some cases, a single tag could be associated with a group of routes rather than with a single route. Thus, much less state is required than would be the case if tags were allocated to individual flows.

In general, a tag switch will try to populate its TIB with incoming and outgoing tags for all reachable routes, allowing all packets to be forwarded by simple label swapping. Tag allocation thus is driven by topology (routing), not traffic. The existence of a FIB entry causes tag allocations, not the arrival of data packets.

The use of tags associated with routes, rather than flows, also means that there is no need to perform flow-classification procedures for all the flows to determine whether to assign a tag to a flow. That simplifies the overall routing scheme and creates a more robust and stable environment.

When tag switching is used to support destination-based routing, it does not eliminate the need for normal network-layer forwarding. First of all, to add a tag to a previously untagged packet requires normal network-layer forwarding. This function can be performed by the first hop router, or by the first router on the path that is able to participate in tag switching. In addition, whenever a tag switch aggregates a set of routes into a single tag and the routes do not share a common next hop, the switch must perform network-layer forwarding for packets carrying that tag. The number of places, however, where routes are aggregated is smaller than the total number of places where forwarding decisions must be made. In addition, aggregation often is applied to a subset of the routes maintained by a tag switch. As a result, a packet usually can be forwarded by using the tag-switching algorithm.

Hierarchical Routing

The IP routing architecture models a network as a collection of routing domains. Within a domain, routing is provided via interior routing (such as OSPF), while routing across domains is provided via exterior routing (such as BGP). All routers within domains that carry transit traffic, however, (such as domains formed by Internet Service Providers) must maintain information provided by exterior routing, not just interior routing.

Tag switching allows the decoupling of interior and exterior routing so that only tag switches at the border of a domain are required to maintain routing information provided by exterior routing. All other switches within the domain maintain routing information provided by the domain's interior routing, which usually is smaller than the exterior-routing information. This, in turn, reduces the routing load on non-border switches and shortens routing convergence time.

To support this functionality, tag switching allows a packet to carry not one, but a set of tags organized as a stack. A tag switch either can swap the tag at the top of the stack, pop the stack, or swap the tag and push one or more tags into the stack. When a packet is forwarded between two (border) tag switches in different domains, the tag stack in the packet contains just one tag.

When a packet is forwarded within a domain, however, the tag stack in the packet contains not one, but two tags (the second tag is pushed by the domain's ingress-border tag switch). The tag at the top of the stack provides packet forwarding to an appropriate egress-border tag switch, while the next tag in the stack provides correct packet forwarding at the egress switch. The stack is popped by either the egress switch or by the penultimate switch (with respect to the egress switch).

The control component used in this scenario is similar to the one used with destination-based routing. The only essential difference is that, in this scenario, the tag-binding information is distributed both among physically adjacent tag switches and among border tag switches within a single domain.

Flexible Routing using Explicit Routes

One of the fundamental properties of destination-based routing is that the only information from a packet that is used to forward the packet is the destination address. Although this property enables highly scalable routing, it also limits the capability to influence the actual paths taken by packets. This limits the ability to evenly distribute traffic among multiple links, taking the load off highly utilized links and shifting it toward less-utilized links. For Internet Service Providers (ISPs) who support different classes of service, destination-based routing also limits their capability to segregate different classes with respect to the links used by these classes. Some of the ISPs today use Frame Relay or ATM to overcome the limitations imposed by destination-based routing. Tag switching, because of the flexible granularity of tags, is able to overcome these limitations without using either Frame Relay or ATM. To provide forwarding along the paths that are different from the paths determined by the destination-based routing, the control component of tag switching allows installation of tag bindings in tag switches that do not correspond to the destination-based routing paths.

Multicast Routing

In a multicast routing environment, multicast routing procedures (such as Protocol-Independent Multicast [PIM]) are responsible for constructing spanning trees, with receivers as leaves. Multicast forwarding is responsible for forwarding multicast packets along these spanning trees.

Tag switches support multicast by utilizing data link layer multicast capabilities, such as those provided by Ethernet. All tag switches that are part of a given multicast tree on a common subnetwork must agree on a common tag to be used for forwarding a multicast packet to all downstream switches on that subnetwork. This way, the packet will be multicast at the data link layer over the subnetwork. Tag switches that belong to a common multicast tree on a common data-link subnetwork agree on the tag switch that is responsible for allocating a tag for the tree. Tag space is partitioned into nonoverlapping regions for all the tag switches connected to a common data-link subnetwork. Each tag switch claims a region of the tag space and announces this region to its neighbors. Conflicts are resolved based on the IP address of the contending switches. Multicast tags are associated with interfaces on a tag switch rather than with a tag switch as a whole. Therefore, the switch maintains TIBs associated with individual interfaces rather than a single TIB for the entire switch. Tags enable the receiving switch to identify both a particular multicast group and the upstream tag switch (the previous hop) that sent the packet.

Two possible ways exist to create binding between tags and multicast trees (routes). For a set of tag switches that share a common data-link subnetwork, the tag switch that is upstream with respect to a particular multicast tree allocates a tag, binds the tag to a multicast route, and then advertises the binding to all the downstream switches on the subnetwork. This method is similar to destination-based upstream tag allocation. One tag switch that is downstream with respect to a particular multicast tree allocates a tag, binds the tag to a multicast route, and advertises the binding to all the switches, both downstream and upstream, on the subnetwork. Usually, the first tag switch to join the group is the one that performs the allocation.

Tag Switching with ATM

Because the tag-switching forwarding paradigm is based on label swapping, as is ATM forwarding, tag-switching technology can be applied to ATM switches by implementing the control component of tag switching. The tag information needed for tag switching can be carried in the ATM VCI field. If two levels of tagging are needed, then the ATM VPI field could be used as well, although the size of the VPI field limits the size of networks in which this would be practical. The VCI field, however, is adequate for most applications of one level of tagging.

To obtain the necessary control information, the switch acts as a peer in network-layer routing protocols, such as OSPF and BGP. Moreover, if the switch must perform routing information aggregation, then to support destination-based unicast routing the switch performs network-layer forwarding for some fraction of the traffic as well. Supporting the destination-based routing function with tag switching on an ATM switch might require the switch to maintain not one, but several tags associated with a route (or a group of routes with the same next hop). This is necessary to avoid the interleaving of packets that arrive from different upstream tag switches but are sent concurrently to the same next hop. Either the downstream tag allocation on demand or the upstream tag allocation scheme could be used for the tag allocation and TIB maintenance procedures with ATM switches.

Therefore, an ATM switch can support tag switching, but at the minimum, it must implement network-layer routing protocols and the tag-switching control component on the switch. It also might need to support some network-layer forwarding.

Implementing tag switching on an ATM switch would simplify integration of ATM switches and routers. An ATM switch capable of tag switching would appear as a router to an adjacent router. That would provide a scalable alternative to the “overlay” model and remove the necessity for ATM addressing, routing, and signalling schemes. Because destination-based forwarding is topology-driven rather than traffic-driven, application of this approach to ATM switches does not involve high call-setup rates, nor does it depend on the longevity of flows.

Implementing tag switching on an ATM switch does not preclude the capability to support a traditional ATM control plane (such as PNNI) on the same switch. The two components, tag switching and the ATM control plane, would operate independently with VPI/VCI space and other resources partitioned so that the components do not interact.

Quality of Service

An important proposed tag-switching capability is quality of service (QoS) support. Two mechanisms are needed to provide a range of QoS to packets passing through a router or a tag switch:

- Classification of packets into different classes
- Handling of packets via appropriate QoS characteristics (such as bandwidth and loss)

Tag switching provides an easy way to mark packets as belonging to a particular class after they have been classified the first time.

Initial classification would be done by using information carried in the network-layer or higher-layer headers. A tag corresponding to the resultant class then would be applied to the packet. Tagged packets could be handled efficiently by the tag-switching routers in their path without needing to be reclassified. The actual packet scheduling and queueing is largely orthogonal: the key point here is that tag switching enables simple logic to be used to find the state that identifies how the packet should be scheduled.

The exact use of tag switching for QoS purposes depends a great deal on how QoS is deployed. If RSVP is used to request a certain QoS for a class of packets, then it would be necessary to allocate a tag corresponding to each RSVP session for which state is installed at a tag switch. This might be done by TDP or by extension of RSVP.

IP Switching

IP switching is a related technology that combines ATM Layer 2 switching with Layer 3 routing. In this way, it is another a type of multi-layer switching. IP switching typically allocates a label per source/destination packet flow. An IP switch processes the initial packets of a flow by passing them to a standard router module that is part of the IP switch.

When an IP switch has seen enough packets go by on a flow to consider it long-lived, the IP switch sets up labels for the flow with its neighboring IP switches or edge routers such that subsequent packets for the flow can be label-switched at high-speed (such as in an ATM switching fabric), bypassing the slower router module. IP switching gateways are responsible for converting packets from non-labeled to labeled formats, and from packet media to ATM.