

EFFICIENT SCALING FOR MULTISERVICE NETWORKS

Flexible Filtering and Industry-Leading Forwarding Performance V2.2 (Revised)

Table of Contents

Executive Summary	1
Introduction: Why We Need Efficient Service Scaling.....	1
The Purpose of the Data Plane	2
Filtering Techniques	3
Content-Driven Lookup	3
ASIC-Driven Memory Search.....	4
Packet Classification in the JUNOS Software Policy Framework	7
Service Control for Operational Efficiency	8
Scaling: Filter Capacity	11
Juniper Networks Filtering Examples	11
Nesting: Reusable Filters	12
Chaining: Forwarding Table Filters	13
Assigning Filters to Interface Groups	14
Filter-Based Accounting and DCU	14
Control Plane – Filter to Loopback	14
Any Field in Any Header	14
Conclusion	15
Appendix A: Header Fields Referenced in Filters	16
Appendix B: Juniper Networks References	17
Application Note	17
Books	17
Documentation	17
White Papers	17
About Juniper Networks.....	18

Table of Figures

Figure 1: Control plane (Routing Engine) and data plane (PFE)	2
Figure 2: High-level PFE functionality	2
Figure 3: Routing system and location of PFE: M320 and T Series	5
Figure 4: Radix tree starting at 192.168.0.0/16 and expanding to /19	5
Figure 5: Locating a range of routes (or ports)	6
Figure 6: Juniper’s filtering and forwarding engines	7
Figure 7: Flowchart logic for nested and chained filters	12

Executive Summary

This paper will be of interest to network operators and managers, as well as executives. It compares and contrasts different architectural approaches for implementing the packet forwarding path in routers.

As traffic volumes rise along with the service demand in IP networks, controlling discrete flows and services is increasingly challenging. To support multiservice networks, providers must be able to establish numerous varieties of tunnels and do so with maximum operational efficiency.

Carriers also require security, traffic control, network visibility and enhanced service-creation capabilities at the edges of their networks, and sophisticated policy in the core. And they need these features to be implemented in hardware so the features do not impact packet-forwarding performance.

In order to support the services in multiservice networks, the data planes of routers must perform stringent packet classification. The basic operations of policy assignment and packet filtering gain in importance as the multiservice networking environment becomes more prominent. The necessary operations boil down to correctly classifying packets; the types of classification that need to be performed include route table lookups and filtering involving multiple fields in the packet.

These operations need to be flexible, and they must be performed at line rate for speeds of (today) up to 100 Gbps with an Internet Mix (IMIX) of packet sizes. Furthermore, the demand for multiservice networking at the core and IP services edge demand the use of packet classifiers that can handle sophisticated pattern searches and complex expression matching.

There are two major architectural approaches to packet classification in high-end routers today. One approach relies heavily on content-addressable memory (CAM); this solution gained popularity because of sheer lookup speed and the capability to invoke many services in a single pass. The other approach emphasizes shared memory lookup algorithms, and is driven by one or more special purpose integrated circuits. This methodology is chosen for its scalability and flexibility in adapting and new services.¹

Juniper Networks® took the latter approach and constructed several highly flexible and extendible Packet Forwarding Engines (PFEs) based on a complex of programmable ASICs. The result is a data plane in all Juniper Networks T Series Core Routers and the high-end Juniper Networks M Series Multiservice Edge Routers—including the M120 and M320 Edge Routers—that provides high security and granular traffic control while handling potentially hundreds of thousands of filtering operations at the highest line rates. This is unparalleled in the industry.

Furthermore, Juniper PFEs are the only forwarding engines in the industry to implement sophisticated filtering policies based on the nesting and chaining operations demanded by modern service providers. Several of the many advantages to service providers that are unique to the Juniper Networks implementation are discussed in this paper.

Introduction: Why We Need Efficient Service Scaling

In order to provide the services in multiservice (voice, video, data, mobile, consumer, business and so on) networks, the data planes of routers must go well beyond destination-based forwarding and source lookups. The burden is on the data plane to provide more, larger and more sophisticated service tunnels—VPNs or MPLS LSPs for instance—and to do so in the most operationally efficient manner.

These requirements vary slightly at the edge and core. In core networks, firewall filters are typically used as one of many solutions that enforce a security policy on the router. Historically, this was true because edge routers could not filter at scale. Policy-based routing is also common at both the network core and edge. These policies can help prevent malicious traffic from entering the network; they need not always be very specific or long to be effective.

In general, policy-based rules at the edge of the network tend to be more stringent. The filtering requirements for security, quality of service (QoS), accounting and other features are usually more specific and the amount of rules are multiplied based on customer applications. A router positioned in the edge or in a collapsed POP solution should have the ability to scale to and match or exceed a real-world network requirement. It must handle such activities as counting packets, filtering, policing, policy-based routing, traffic sampling, QoS, and many other classification tasks that involve table lookups and actions to perform field and route matching.

¹Additional details on both of these methods are described in the *Filtering Techniques* section and in the section on *Packet Classification in the JUNOS Framework*.

For example, traffic policing allows service providers to classify packets and assign different packet flows to different policing thresholds. Sampling supports the ability to aggregate sampled traffic and send flow information to a remote host or management station. Firewall filters may include alert actions that cause packet-matching filter conditions to be logged into a system log file that can then be sent to a system log server.

Other useful innovations include a source class usage and a destination class usage (SCU/DCU) feature that supports advanced billing and accounting capabilities by allowing providers to maintain packet counts based on the ingress and egress points of traffic transmitted through their networks. Finally, filter-based forwarding (FBF) supports provider open access requirements by providing the ability to forward a packet from a specific physical or logical interface based on the packet's source address.

The Purpose of the Data Plane

A modern router is divided into a control plane and a data plane. The functionality of the control plane includes the routing protocol software in the Routing Engine; this includes complex control functions. There is a well-defined interface between the control plane (Routing Engine) and data plane (PFE).

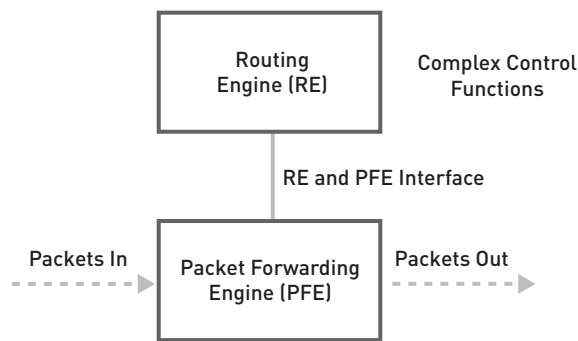


Figure 1: Control plane (Routing Engine) and data plane (PFE)

The data plane of the router handles the packet processing from ingress to egress. The components involved in this operation are the PFE and the switch fabric. On a high level, the functions of a PFE are illustrated in the following diagram.

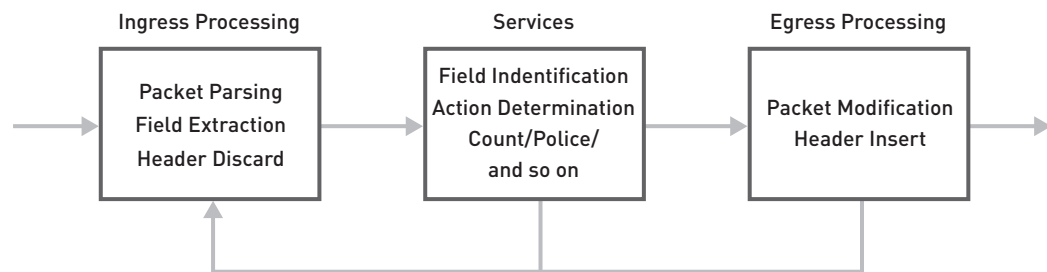


Figure 2: High-level PFE functionality

The PFE determines data packet forwarding through the router. Within a router's PFEs², services can be defined in terms of two linked operations: *identification* (of, for instance, one or more fields within a packet, a route prefix or a range of Layer 4 ports) and an *action* (such as count, drop, forward, police or mark QoS).

The most commonly used identification methods involve firewall filtering—matching on source and destination IP addresses or ports—but there are many others as well. The following table shows classification methods at different layers of the OSI model.³

² Most large routers have multiple PFEs, perhaps one or more per line card.

³ For a list of all the fields in the packet that may be used for packet classification on Juniper Networks routers, see *Appendix A: Header Fields Referenced in Filters*.

Table 1: Packet Classification Methods

OSI LAYER	APPLICATION	MULTI FIELD	MATCH TYPE (IDENTIFICATION)	DESCRIPTION
2	Packet Switching	No	Exact	MAC (specified) defines forwarding
2	CoS: 802.1p	No	Exact	IP precedence bit setting changes queue
2.5	MPLS	No	Exact	Label switch identifies next hop
2.5	MPLS EXP	No	Exact	MPLS QoS bit setting changes queue
3	Packet Routing	No	Longest Prefix	Destination IP network is identified
3	IP Options	Yes	Exact	IP Options field matched for security
3	Stateless firewall	Yes	Exact/Longest Prefix	Filter based on source or destination
4	Stateful firewall	Yes	Prefix or Range	Filter based on an IP flow
4	DiffServ	Yes	Prefix or Range	ToS bit setting changes queue
4	IntServ Flow	Yes	Exact	Prioritization based on IP and port addresses
4	Hybrid QoS	Yes	Exact and/or Range	Combination of CoS, ToS and EXP
7	Load Balancing	Yes	Exact or Prefix	Packets with payload type are redirected
7	Intrusion Detection	Yes	Signature Scanning	Redirect packets if string in payload

In general, the simplest forwarding operations require matching on only a single field. Many packets will of course be analyzed and classified based on multiple criteria. Filtering operations—which may include firewalls, VPNs and complex QoS operations (such as per-VLAN or per-VPN QoS)—need to implement policies based upon Layers 2 through 4 in the packet header. Other classification schemes (Layer 7) are content based; intrusion detection and load balancing require scanning the packet for particular values before determining an action.

The performance of packet analysis and classification on a router's data plane directly affects the router's overall performance. For example, filtering based upon QoS bits at multiple layers of the OSI model, as well as route prefixes and port ranges, can quickly exhaust a too rigid PFE. A security-related example is the prevention of denial of service (DoS) attacks, which often requires large filter lists, and which can be much more easily mitigated through a chained operation that does not require the whole set of filtering terms to be reworked.

Filtering Techniques

There are two major filtering techniques in use for a PFE. One common method is to use special-purpose classification devices such as CAMs/Ternary CAMs (TCAMs). These memories have strong performance features. The other major technique performs firewall filter actions using similar algorithms to those used in route lookups. In order to maximize the efficiency and scale of these operations, special-purpose ASICs and sophisticated algorithms and data structures are used.

Content-Driven Lookup

When a specialized packet-matching structure (such as TCAM) is used for implementing services, the memory is segmented by service features needing identification. When a packet arrives on an ingress interface, it is scanned in parallel in all TCAM segments. Thus, all service features are evaluated simultaneously. Those segments that positively identify the packet then flag it for further processing. A network processor takes the results and executes the needed actions, in a fixed order that cannot be modified by the operator.

As with any architectural choice, there are trade-offs that must be considered; the speed of content-driven lookups comes with some costs. These costs include limited control over service sequences and combinations—ordering of services is fixed and (in all implementations to date) services cannot be easily combined or repeated; a content-driven data plane cannot be programmed to count packets, drop some of the packets and then count again.⁴ Other costs include power and expense.

⁴This is demonstrated in the section below on *Service Control for Operational Efficiency*.

Because content-addressable memory involves a bit-based lookup, multiple entries are needed where IP routes or Layer 4 port ranges cross bit boundaries. The same phenomenon is found with any kind of summarization that crosses bit boundaries, such as summarizing contiguous networks using variable-length subnet mask (VLSM).

With CAM implementations, firewall filters are typically optimized (compiled) before being downloaded to CAM. This process has a few drawbacks. One is that compilation causes the filter lines to no longer be associated with unique or single CAM cells.

Because of this, providers cannot count packets per firewall filter term. Also, non-contiguous matches (caused by the crossing of bit boundaries) require multiple CAM cells to be allocated for a single match. This process is known as a “CAM explosion” and it is readily demonstrated.⁵ As a result, these range-matching filters may deplete CAM space prematurely and cause either the configuration or the forwarding process to fail.

Currently, most vendors implement a hybrid architecture, where route lookups are done via memory search, and services are programmed in TCAMs. This architecture became popular because it avoids the challenge of making a lookup engine fast enough to process routes and services concurrently. The main trade-off with TCAM-based services is the lack of flexibility and programmability inherent in memory-based architectures (discussed next). A second trade-off, which gets increasingly visible as energy efficiency becomes more critical, is extensive power consumption of the TCAM chips. When planning for energy-efficient networking, TCAMs are less attractive than alternatives.

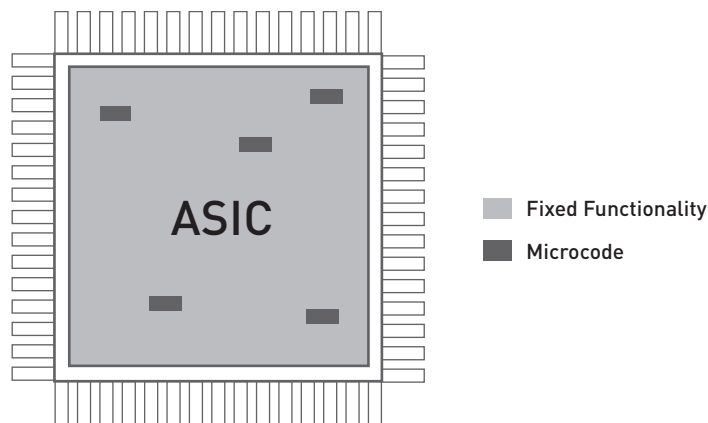
ASIC-Driven Memory Search

The other approach to packet classification uses more conventional memory architectures (such as SRAM for instance) and drives this memory from ASICs that store information and search for it using sophisticated data structures. Identification criteria involve matching against a particular term before taking action.⁶

The flexibility comes from the correct choice of which functions should be implemented in hardware (never to change for the life of the ASIC) and which should be implemented with microcode. Microcode is the mechanism that allows ASICs to provide the flexibility that is needed to extend hardware-based router longevity to support deployable lifetimes of many years.⁷ A large amount of hardware coupled with a small amount of microcode can provide a tremendous amount of flexibility.

It’s also interesting to note that memory-driven stateless services have long been considered impossible to build from the cost perspective, because they require packet lookup budgets far beyond what could be practically achieved. Juniper Networks Internet Processor II became the world’s first ASIC capable of running flexible services at 20 G+ line rate, and subsequent Juniper products were built around the same methodology.

ASICs Have Fixed Functionality and Microcode



⁵ Refer to vendor documentation on restrictions for TCAM merge algorithms.

⁶ Historically, memory search algorithms were the first to emerge for route lookups because of their relative ease of implementation on general-purpose CPUs. Realizing the need for lookups at much higher speeds, Juniper pioneered the industry with ASIC-based route search at line rate on the original M40 router and related IP processor. The same approach was later extended to PFE-based services, which form service programs similar to the route lookup routines.

⁷ Many Juniper customers have deployed routers in excess of 10 years, as it is Juniper’s philosophy to build next-generation networks with the most incremental upgrades possible. Juniper is the only vendor that fully supports the most expensive hardware parts (PICs) built since 1998 with modern feature sets on the new platforms.

There are a number of these ASICs that may comprise the core chipset of the PFE. In the Juniper implementation, hardware is developed using the highest-density chips that silicon technology allows us to build.⁸ Juniper’s use of highly integrated silicon lowers costs, increases performance, increases reliability and lowers power consumption. The PFE in M Series, T Series, and Juniper Networks MX Series Ethernet Services Routers are built using a core chipset that consists of several chips, embedded software that controls these chips, and interfaces between the embedded software and the core chipset. Such a system consists of four major components: PICs, PFEs, the switch fabric and one or more Routing Engines.

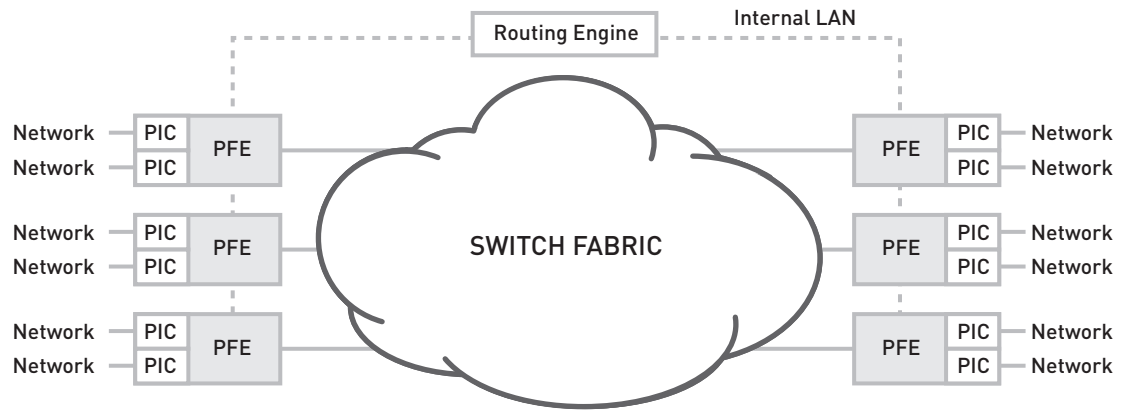


Figure 3: Routing system and location of PFE: M320 and T Series

The PFE implementation—located in the FPCs that house the PICs—is based on a number of sophisticated data structures. Among the major primitives are tree lookups, table lookups and filters. No other vendor has been able to build a device with lookup speeds approaching the algorithms implemented by Juniper.⁹

The tree lookup is based on a device for binary number matching called a radix tree. A radix tree can be a graphical representation of the binary numbers that make up, for instance, route prefixes and port ranges. For instance, the following radix tree starts at 192.168.0.0/16 and expands three levels to eight subnetworks with longer matches using a /19 prefix.¹⁰

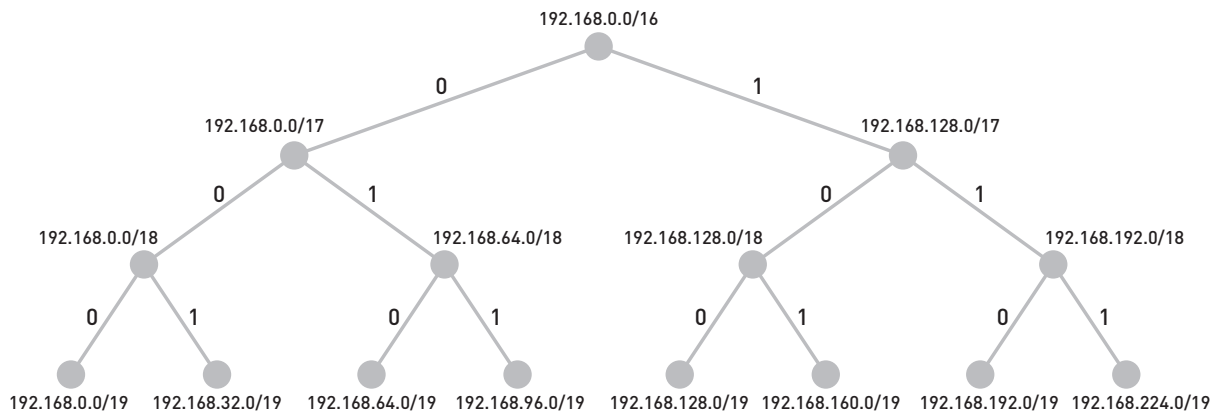


Figure 4: Radix tree starting at 192.168.0.0/16 and expanding to /19

This device allows for very complex and controlled services chains. The advantage of the approach is that operations on a range of ports or a set of routes can be based on a pointer from a specific node of the tree. In the following figure, all of the routes that are more specific than 192.168.0.0/16 are shown in the highlighted section.

⁸ For example, current technology level is 90-nm fabrication. We are expecting to move to 65 nm and better design processes as soon as they becomes practical for building network-related silicon.

⁹ Note also that the 50-Gbps T Series PFE is currently the newest, and most flexible and highest performing PFE on the market.

¹⁰ For more information, see *JNCIA: Juniper Networks Certified Internet Associate Study Guide*, by Joseph M. Soricelli et al, Sybex, 2003.

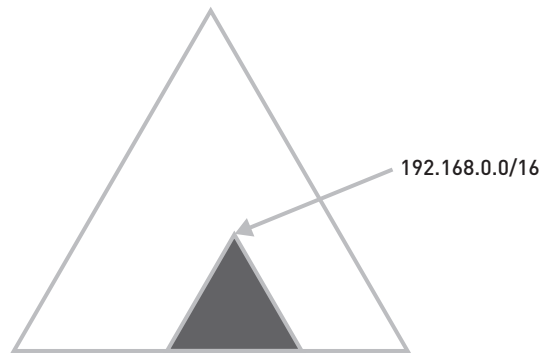


Figure 5: Locating a range of routes (or ports)

Because of this ease of identification, there is very little processing power required to connect multiple identification steps with multiple actions. For example, the following construct is very simple to implement in Juniper Networks routers.

```
.....
Identification -> action -> further identification -> action-2
.....
```

This flexibility is due to the ability to chain together the different primitives—tree lookups, table lookups and filters—in any order. Embedded software is used to program and control the basic structure and sequence of the packet processing primitives supported by each release of Juniper Networks JUNOS® Software. The embedded software allows engineers to add a primitive to the chain, to remove a primitive from the chain and to execute these primitives in any combination, in any order, on virtually any type of data packet.

It is very easy for service providers to benefit from this ability. They simply upgrade their systems to a new version of JUNOS, and the new software features are immediately supported by Juniper’s hardware-based forwarding engine.

Furthermore, there is more flexibility with the syntax and structure of packet classification, and the combinations are essentially a high-level programming language. The flow in this language is based on the identification/action pairs, and the potential list of actions list is very long. The following is a partial list just to give an idea:

- Policing and multi-field classification of traffic
- Counting packets for DoS mitigation or accounting purposes
- Sampling packets for offline analysis or lawful intercept
- Dropping packets as part of attack mitigation or rate limiting
- Chaining to a different forwarding table (forwarding table)
- Chaining to a different MPLS VPN (Virtual Routing and Forwarding instance or VRF)
- Redirecting to mirror port or services engine

This flexibility provides network operators with ultimate control over the classification and processing of packets from ingress to egress in the router. Other benefits include scalability, ease of matching and ease of counter implementation. These are discussed in more detail in the next section.

Packet Classification in the JUNOS Software Policy Framework

The JUNOS Software policy framework controls the flow of packets into and out of the router. The policy framework has two related components:

- **Routing policy:** Allows providers to control the routing information between the routing protocols and the routing tables and between the routing tables and the forwarding table
- **Firewall filter policy:** Allows providers to control packets transiting the router to a network destination and packets destined for and sent by the router

The functionality of routing policy and firewall filtering policy are extremely similar, which is a design choice made because of the relatedness of the tasks behind routing protocol and table management (in the Routing Engines on the control plane) and the tasks affecting which packets are accepted and sent between interfaces on the router (in the PFEs on the data plane).

Indeed, the very phrase *firewall filter* policy is used here to emphasize that a firewall filter is in fact a policy and shares fundamental similarities with a routing policy. A *firewall filter term*¹¹ is a classification technique that includes policy-based rules and can be applied to individual packets when a match occurs. The firewall filter term is a pairing of the identification (match) and action to be taken based on the identification.

JUNOS firewall filter capabilities are unique in two major areas:

- Able to contain multiple matches and conditional actions; no competing implementation performs nested and chained filtering (described next)
- Huge scalability; content-driven implementations allow for a mere fraction of the total firewall filter terms supported on Juniper routers without performance degradation

The following figure illustrates how a Juniper PFE stores multiple tables and filters, each of which can be flexibly accessed for lookups or for applications to individual interfaces. The three major primitives illustrated here are tree lookup, table lookup and filters.

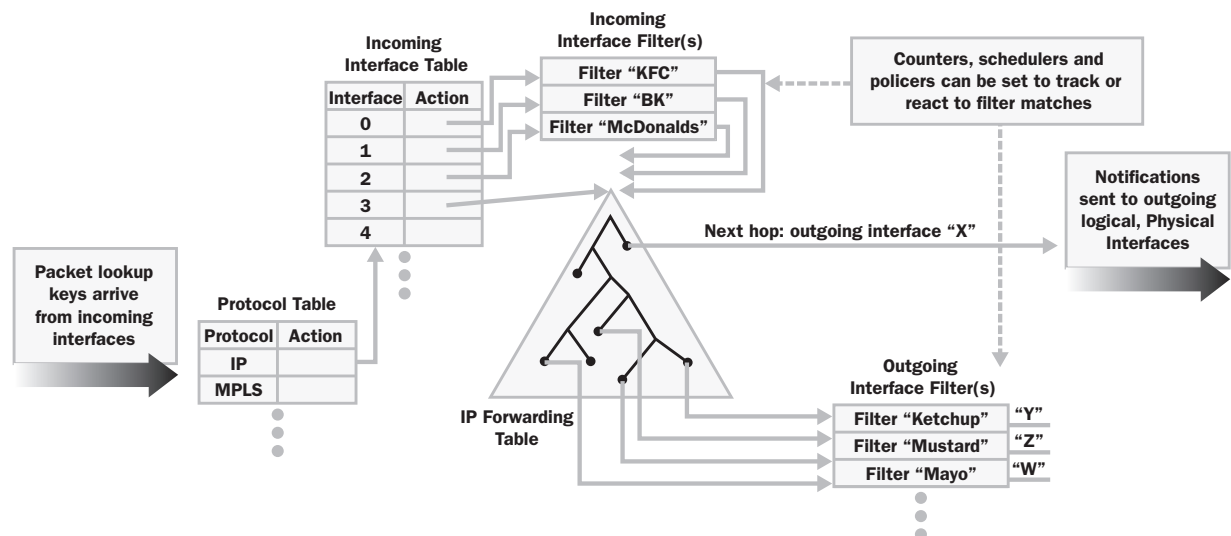


Figure 6: Juniper's filtering and forwarding engines

When packets arrive on a particular interface, the input interface table-lookup primitive determines if the interface is active, and assuming it is, the notification is passed to the tree-lookup primitive to perform a longest-match prefix lookup in the forwarding table. Based on the information returned from the route lookup, an output filter may be performed against the packet notification. If the output filter accepts the packet notification, the notification is sent to the packet transmission queue of the output interface.

This figure is an abstraction of course. There are many more tables, many different data structures and algorithms, and multiple ways to invoke the referenced services. But the uniqueness of this approach is the ability to chain together multiple functional building blocks (that is, route lookup, table lookup and filtering).

¹¹ Other network equipment vendors may refer to this construct as an access control entry.

For example, IP traffic can be directed by the incoming protocol table to a second incoming interface table. The operator can configure various “next actions” for each interface. The next action could be to apply a filter or it could be to send the packet directly to the IP forwarding table (FIB).¹² The next action could also be to redirect the packet directly to a particular interface. After a lookup is performed, filters can be configured and applied for particular outgoing interfaces to filter traffic destined for particular next hops, before sending notifications to outgoing interfaces. Counters can be configured to track the number of matches for each filter.

Individual filters do not need to be duplicated for different interfaces. They can be stored in memory once and applied to selected interfaces or a range of interfaces, and they can be applied in different orders based on externally applied conditions. This feature, which is not available on the content-driven PFE architectures of competitors, reduces the number of entries that need to be configured and processed on the router.

Service Control for Operational Efficiency

One easy way to understand the benefits of the programmable PFE behavior is to look at a configuration typical of routers using content-based lookups. These routers normally have multiple services defined separately and applied to the target one by one, in line with TCAM segmentation.

Here is an example that defines filtering, policing, sampling and policy routing on an interface of a router with TCAM-based services.

```
.....  
!  
! Interface Configuration  
!  
interface XX  
  ip address 192.2.1.1 255.255.255.0  
  route-map Rmap  
  service-policy QoS_Policy_In input  
  ip access-list 100 in  
  ip sampling-policy Sample  
!  
! Route Map  
!  
route-map Rmap permit 10  
match acl 101  
set ip next-hop 10.0.0.1  
!  
! Class Map  
!  
class-map match-all Class1  
match access-group 102  
!  
! Policy Map  
!  
policy-map QoS_Policy_In  
class Class1
```

¹² Another good example is a forwarding table filter, something unique to the Juniper PFE architecture.

```

police 10000000 4470 4470 conform-action transmit exceed-action drop
!
! Class Map
!
class-map match-all Class2
match access-group 103
!
! Sampling
!
sampling-policy Sample
class Class2
!
! Access Lists (Equivalent to Firewall Filter Terms)
!
access-list 100 deny udp 192.161.41.49 0.0.0.0 any access-list 100 permit ip any any
access-list 101 permit ip 192.73.81.0 0.0.0.255 any access-list 101 deny ip any any
access-list 102 permit ip 192.73.0.0 0.0.255.255 any access-list 102 deny ip any any
access-list 103 permit ip any any
!

```

Although all services in the previous example use filters for packet classification, they do not form a contiguous policy. Instead, packets arriving on the interface are subjected to discrete actions applied in a non-obvious order. For example, it is impossible to determine whether sampling or policy routing will work before or after the policer action. It is also impossible to count packets passing through every services gate. This is because non-contiguous services policies (in these competitive solutions) do not allow for actions to be freely combined and repeated. To enforce a packet passing through multiple actions (that is, policing, tagging, filtering and sampling) in a specified order, all of the actions would require explicit, separate, non-reusable configurations. This is operationally untenable.

The previous pseudo-configuration is much easier to program and control in a structured language (JUNOS Software), where packet flow and actions are explicitly defined in one services chain.

```

filter Input {
    term Reject-traffic {
        from {
            address {
                192.161.41.49/32;
            }
            protocol udp;
        }
        then {
            count deleted-counter1;
            discard;
        }
    }
}

```

```
term Policy-routing {
  from {
    address {
      192.73.81/24;
    }
  }
  then {
    count policy-forwarded-counter2;
    routing-instance isp1-route-table;
  }
}
term Police {
  from {
    address {
      192.73/16;
    }
  }
  protocol udp;
  then {
    count policed-counter3;
    policer policer-1;
    accept;
  }
}
term Accept {
  then {
    accept;
  }
}
policer policer-1 {
  if-exceeding {
    bandwidth-limit 10m;
    burst-size-limit 1m;
  }
  then {
    discard;
  }
}
```

In JUNOS Software, configuring services in a single chain allows for the very granular control over the packet life. For example, adding “accept” to the action block in the term “Policy-routing” will explicitly define that this traffic is not rate-limited. Otherwise, it is.

Packet-matching conditions could be built from any bit fields or any tags (that is, Destination Class Usage or interface group) applied internally on the router. Range matches are allowed in all fields, and multiple match conditions may apply to a single term.

Likewise, match actions could be anything the PFE can handle and may be repeated in any arbitrary order. This type of configuration style is not only more flexible, but also less prone to complex programming errors stemming from interaction between services.

Scaling: Filter Capacity

The other area where M Series and T Series routers stand alone is scalability. Fundamentally, scalability has to do with the ability of the PFE to process packets. The filtering activities include—in addition to security features—multicast, IP routing (table lookups and outbound interface assignments), edge aggregation, label assignment and swapping, and so forth.

Core network devices must be able to filter traffic and allow carriers to set policy. The greater the number of firewall filter terms that an interface can support, the greater control that a carrier has over the traffic traversing the core. Firewall filter term handling is thus an important test of scalability.

T Series Core Routers and M Series Multiservice Edge Routers easily accommodate hundreds of thousands of firewall filter entries while forwarding traffic at line rate. These configurations have been tested on numerous packet forward engines.

Firewall filters are expected to introduce additional latency because of the additional lookup in the forwarding path. As the size of the filter increases, the time taken to traverse it increases. However, in M Series and T Series routers, only extremely low packet sizes (for example, entire streams of 40-byte packets, never seen in real networks) are affected by very large filters.¹³

With TCAM-based implementations, one packet forwarding entry is needed to accept or deny one IP prefix. Furthermore, there is only a limited concept of ranges with firewall filter terms. Multiple entries are always required to satisfy ranging of IP addresses or TCP/UDP ports, as long as the range crosses bit boundaries.¹⁴

Thus, operators can quickly exhaust the maximum allocation of firewall filter terms. This is the “CAM explosion” phenomenon previously described. Once the maximum number of firewall filter terms supported in hardware is exceeded, the processing of these filter terms ceases or reverts to the software forwarding path.

Juniper Networks Filtering Examples

The following sections provide examples of features that are only possible using Juniper Networks firewall filters. These features allow the router to handle many more services than the competition and to do so with much greater operational efficiency.

For more detailed configuration examples of these features, see *Appendix B: Juniper Networks References*.¹⁵

Overview of Nested and Chained Filters

Nested and chained filters are unique capabilities on the M Series and T Series routers. With the ability to nest a firewall filter, it can be reused in several places. Also, filters can be chained in features called forwarding table filtering. The following diagram shows the logical flow of these features.

¹³ For test results illustrating this scale, contact your Sales Engineer (SE) and refer your SE to this paper and to the Competitive Intelligence team.

¹⁴ For detailed examples of test cases and scaling deficiencies in content-based lookup implementations, contact your Sales Engineer (SE) and refer your SE to this paper.

¹⁵ All of these examples are discussed in more detail in the *JUNOS Software Release 9.1 Policy Framework Configuration Guide*.

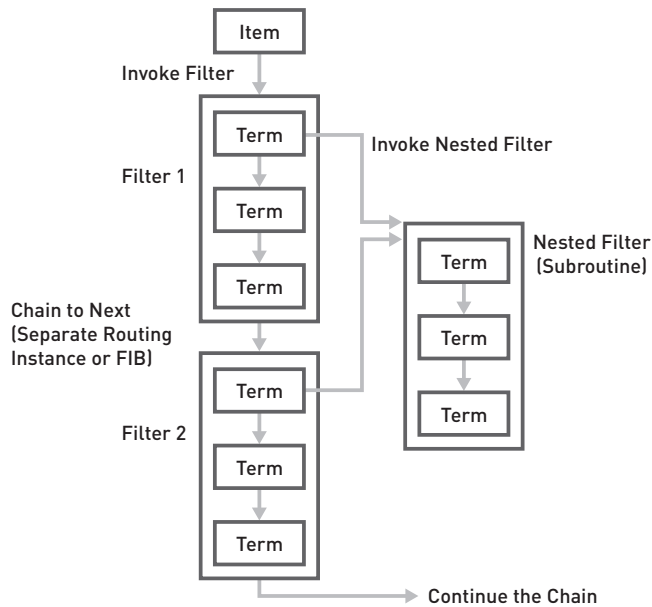


Figure 7: Flowchart logic for nested and chained filters

The nesting capability allows a number of applications. One example is to have a separate filter check IP address prefixes after protocol redistribution; another is to reject unallocated (Bogon) prefix blocks. A third usage of this feature might be to police (rate limit) after a traffic count on an interface reaches a specified threshold (this count may be different for different interfaces, logical or physical). In all cases, a separate filter is created and invoked when needed.

The main application for chaining is in a feature called forwarding table filter (FTF). With FTF, operators can specify filters for individual routing instances such as an MPLS VRF, the Internet and the control plane of the router.

Nesting: Reusable Filters

Nested filters minimize the work needed to configure terms common to numerous filters. One filter can be configured with the common desired terms, and they can be applied to other filters. To make changes to the common desired terms, term modifications need to be made only to the filter with the common terms instead of changing terms on every filter.

To configure a filter within a filter, include the filter statement within the firewall filter term:

```
.....
term term-name {
    filter filter-name;
}
.....
```

Define a filter common-filter and configure it into two separate filters:

```
.....
[edit]
firewall {
    filter common-filter {
        term t1 {
            from {
                protocol udp;
                port tftp;
            }
            then {
                log;
                discard;
            }
        }
    }
}
filter filter1 {
.....
```

```

    term term1 {
        filter common-filter;
    }
}
filter filter2 {
    term term1 {
        filter common-filter;
    }
}
}

```

.....

The common filter can then be applied anywhere.

Chaining: Forwarding Table Filters

Chaining is performed by applying a filter to a forwarding table. This feature is called a forwarding table filter (FTF). An FTF controls which packets the router accepts and performs a forwarding table lookup, thereby controlling which packets the router forwards on the interfaces. All packets are subjected to the input forwarding table filter that applies to the forwarding table.

Forwarding table filters allow providers to apply filters to routing instances to better control their network services. As one example, providers can have separate forwarding filters for:

- Access networks
- MPLS VPNs
- IPsec VPNs
- Internet usage
- Router's control plane
- Special forwarding table for when there is DoS

Instead of being applied to interfaces, FTFs are applied to routing instances (forwarding tables). For example, when the router receives a packet, it determines where to forward the packet by searching in a forwarding table—which is associated with the VPN on which the packet will be sent—for the best route to the destination. The router then forwards the packet toward its destination through the appropriate interface. This feature is also a key enabler for advanced BGP-based distributed denial of service (DDoS) protection techniques (draft-marques-idr-flow-spec).

To apply an FTF to a VPN VRF table, providers first create the address family (for example, IPv4, MPLS and so on), then the filter terms, match conditions and actions. Following that, providers edit the routing instance (forwarding table) on which they will apply the filter.

.....

```

[edit]
routing-instance routing-instance-name {
    instance-type forwarding;
    forwarding-options {
        family family-name {
            filter {
                input filter-name;
            }
        }
    }
}
}

```

.....

Assigning Filters to Interface Groups

The use of interface groups allows providers to have unique counters for each interface—this is an extremely useful feature in parrying DoS attacks. Juniper Networks is the only router vendor that enables this.

When an interface group is defined, packets received on that interface are tagged as being part of the group. Providers then can match these packets using the interface-group match statement.

To define an interface to be part of an interface group, providers include the group statement. Providers then create a filter that contains an interface group. Finally, providers assign one or more interfaces to the interface group referenced in the filter, and apply the filter that contains an interface group.

Filter-Based Accounting and DCU

Juniper Networks has a number of billing and reporting features to enhance the collection of accounting information. Filter-based accounting allows providers to configure customized packet filters to identify and then count user-defined traffic flows. Destination class usage (DCU) allows providers to align their accounting policy with their routing policy and to have it dynamically adapt to rapidly changing routing environments. These are both implemented in hardware.

Juniper's implementation of source class usage (SCU) and DCU is unique. It is based on class-based filter conditions. Class-based filter conditions match packet fields based on a source class or destination class.

A source class is a set of source prefixes grouped together and given a class name. A destination class is a set of destination prefixes grouped together and given a class name.

The source class is specified in the following way:

```
.....  
[edit firewall filter inet filter-name term term-name]  
  
from {  
  source-class class-name;  
}
```

The destination class is specified in the following way:

```
.....  
[edit firewall filter inet filter-name term term-name]  
  
from {  
  destination-class class-name;  
}
```

Providers can specify a source class or destination class for an output firewall filter. Although providers can specify a source class and destination class for an input firewall filter, the counters are incremented only if the firewall filter is applied on the output interface.

The class-based filter match condition works only for output filters because the SCU and DCU are determined after route lookup.

Control Plane – Filter to Loopback

Providers can also apply one input or output firewall filter to the routing platform's loopback interface, which is the interface to the Routing Engine (on all routing platforms). This allows providers to filter local packets received by or forwarded from the Routing Engine.

Any Field in Any Header

The table in "Appendix A: Header Fields Represented in Filters" shows all the fields that can be used in firewall filtering terms. The entries in bold are unique to Juniper Networks routers. For example, only Juniper Networks routers can filter on forwarding class.

Conclusion

Many services are built on a foundation of filtering capability. Examples include counters, filter classifiers, policy-based forwarding table assignments, sampling and many other features. To support these services at maximum scale, providers need a solution that does not use fixed algorithms but rather enjoys the flexibility of table searches in memory. Juniper Networks provides this scalability and flexibility with its unique data plane.

For instance, on Juniper Networks M Series, MX Series and T Series routers, providers can count packets, do further filtering, sample, count again and move to another VRF. This sequence is impossible on other vendors' platforms, as are filter chaining and multiple matches in a term. For other vendors, even such basic constructs as repeated counters are problematic because of the necessity to collapse TCAM entries and thus be unable to match firewall filter terms one by one. By contrast, the order of services and degree of control offered on M120, M320 and T Series routers is extensive.

For any viable network implementation, Juniper's memory search algorithm scales significantly better than simple table lookup methods. The limitations of TCAM-based services are in the areas of flexibility and programmability, as well as cost-effective scaling.

Juniper's hardware technology group answers these limitations by designing ASICs with lookup speeds fast enough to process routes and services at wire rate and with great flexibility—on average, every generation of Juniper IP packet processors is two or more times faster than any competing design.

Appendix A: Header Fields Referenced in Filters

The following table shows header fields that can be referenced in filters. The ones that only apply to JUNOS Software (not available in any other vendors' OS) are in bold.

Table 2: Header Fields Referenced in Filters

ADDRESS	MATCH IP SOURCE OR DESTINATION ADDRESS
ah-spi	Match IPsec AH SPI Value
ah-spi-except	Do not match IPsec AH SPI value
apply-groups	Groups from which to inherit configuration data
apply-groups-except	Don't inherit configuration data from these groups
destination-address	Match IP destination address
destination-class	Match destination class
destination-class-except	Do not match destination class
destination-port	Match TCP/UDP destination port
destination-port-except	Do not match TCP/UDP destination port
destination-prefix-list	Match IP destination prefixes in named list
dscp	Match Differentiated Services (DiffServ) code point
dscp-except	Do not match Differentiated Services (DiffServ) code point
esp-spi	Match IPsec ESP SPI value
esp-spi-except	Do not match IPsec ESP SPI value
first-fragment	Match if packet is the first fragment
forwarding-class	Match forwarding class
forwarding-class-except	Do not match forwarding class
fragment-flags	Match fragment flags
fragment-offset	Match fragment offset
fragment-offset-except	Do not match fragment offset
icmp-code	Match ICMP message code
icmp-code-except	Do not match ICMP message code
icmp-type	Match ICMP message type
icmp-type-except	Do not match ICMP message type
interface	Match interface name
interface-group	Match interface group
interface-group-except	Do not match interface group
interface-set	Match interface in set
ip-options	Match IP options
ip-options-except	Do not match IP options
is-fragment	Match if packet is a fragment
packet-length	Match packet length
packet-length-except	Do not match packet length
port	Match TCP/UDP source or destination port
port-except	Do not match TCP/UDP source or destination port
precedence	Match IP precedence value

ADDRESS	MATCH IP SOURCE OR DESTINATION ADDRESS
precedence-except	Do not match IP precedence value
prefix-list	Match IP source or destination prefixes in named list
protocol	Match IP protocol type
protocol-except	Do not match IP protocol type
source-address	Match IP source address
source-class	Match source class
source-class-except	Do not match source class
source-port	Match TCP/UDP source port
source-port-except	Do not match TCP/UDP source port
source-prefix-list	Match IP source prefixes in named list
tcp-established	Match packet of an established TCP connection
tcp-flags	Match TCP flags
tcp-initial	Match initial packet of a TCP connection
ttl type	Match for IPv4 TTL, useful for Generalized TTL Security Mechanism (RFC 3682)

Appendix B: Juniper Networks References

The following references contain more information on the features and functionality described in this white paper. Juniper's prominence in the multiservice edge market, which has always included Ethernet, is discussed in the following documents.

Application Note

Filter-Based Forwarding

An illustration of the use of filters to allow routers to differentiate traffic streams on interfaces directing them to the appropriate routing instance

<http://www.juniper.net/us/en/local/pdf/app-notes/3500136-en.pdf>

Books

JNCIA: Juniper Networks Certified Internet Associate Study Guide, by Joseph M. Soricelli et al, Sybex, 2003.

JUNOS Cookbook: Time-Saving Techniques for JUNOS Software Configuration, by Aviva Garrett, O'Reilly, 2006.

Documentation

JUNOS® Software Release 9.1 Policy Framework Configuration Guide

<http://www.juniper.net/techpubs/software/junos/junos91/swconfig-policy/swconfig-policy.pdf>

White Papers

Securing Service Provider Networks

A white paper on advanced security techniques for securing infrastructure or providing managed security services

<http://www.juniper.net/us/en/local/pdf/whitepapers/2000180-en.pdf>

About Juniper Networks

Juniper Networks, Inc. is the leader in high-performance networking. Juniper offers a high-performance network infrastructure that creates a responsive and trusted environment for accelerating the deployment of services and applications over a single network. This fuels high-performance businesses. Additional information can be found at www.juniper.net.

Corporate And Sales Headquarters

Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, CA 94089 USA
Phone: 888.JUNIPER
(888.586.4737)
or 408.745.2000
Fax: 408.745.2100

APAC Headquarters

Juniper Networks (Hong Kong)
26/F, Cityplaza One
1111 King's Road
Taikoo Shing, Hong Kong
Phone: 852.2332.3636
Fax: 852.2574.7803

EMEA Headquarters

Juniper Networks Ireland
Airside Business Park
Swords, County Dublin,
Ireland
Phone: 35.31.8903.600
Fax: 35.31.8903.601

Copyright 2009 Juniper Networks, Inc. All rights reserved. Juniper Networks, the Juniper Networks logo, JUNOS, NetScreen, and ScreenOS are registered trademarks of Juniper Networks, Inc. in the United States and other countries. "Engineered for the network ahead" and JUNOSe are trademarks of Juniper Networks, Inc. All other trademarks, service marks, registered marks, or registered service marks are the property of their respective owners. Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

To purchase Juniper Networks solutions, please contact your Juniper Networks representative at 1-866-298-6428 or authorized reseller.

