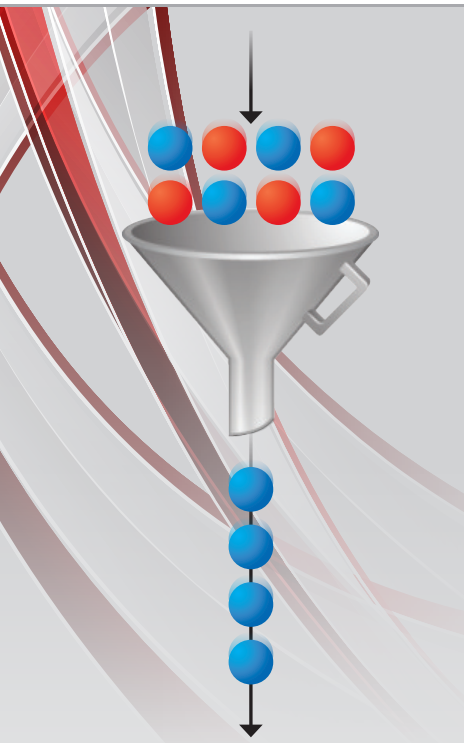
Creating Effective Packet Filters with APCON Intelligent Data Traffic Management Solutions

Technical Brief

Introduction – The Need for Effective Filters

With the growing volume of data passing through packet-switched networks, effectively monitoring production network traffic becomes more challenging. Network administrators are tasked with monitoring a wide array of increasingly complex traffic streams driven by widespread adoption of broadband Internet service, smart phones, and tablets. As a result, the requirement for effective packet filtering in the network monitoring system has become critical to avoid overloading monitoring and analysis tools. Effective and efficient packet filtering becomes a key tool in this area, and APCON Professional Services leads the industry in practical filtering experience.



Filtering is also an important data security tool – recognizing the need to route only required data to individual monitoring tools for tightly focused analysis. Limiting data distribution helps maintain confidentiality and limits the potential for unauthorized disclosure.

This paper details general guidelines and effective practices for implementing filters on APCON IntellaFlex™ network monitoring switch products. It is intended to assist network engineers in efficiently utilizing the filtering features in the switch firmware. By combining packet filtering with packet aggregation capabilities, engineers can more precisely route data to ensure best utilization of expensive monitoring, analysis and recording tools.

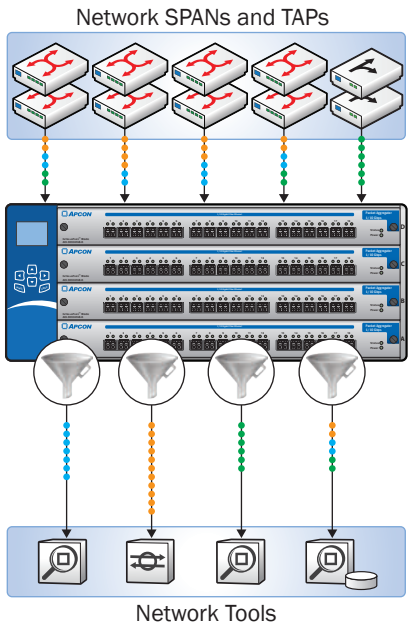
Filter Rules and Effective Practice

Simply described, a filter is a set of rules to process traffic. When a network engineer creates a filter, the rule-generation system expands the filter specifications into a much larger set of underlying computing rules that implement the specified action. In any switch, the total number of rules that can be created depends on the number of ports and packet processors on each blade in the switch. On most APCON IntellaFlex blades, up to 4096 rules are allowed per filter. As another example, a single filter may consist of up to 1536 rules on the 18-port 10G (ACI-3030-E18-6) blade.

In general, we recommend that engineers use an approach that reduces the number of rules generated by the filter, conserving the compute resources required to implement that filter.

One of the most common tasks for filtering is to pass or drop traffic based on a range of IP or MAC addresses. Setting range filters using an upper and lower limit generates many rules, making the filter slower to load. Filters that are very complex or include wide ranges may take several minutes to load before they take effect.

Because a simple-looking statement can be expanded by the rule-generation system into multiple rules, it is possible to create a filter that exceeds the blade's rule limit. However, by simply reversing the logic of a filter (such as changing the specification from pass to drop, or from drop to pass), the final filter can sometimes be made much smaller.



Effective packet filtering in an APCON IntellaPatch® switch with IntellaFlex blades allows you to direct optimized data streams to any monitoring device on your network.

Limitations for Effective Filtering

It is important to understand the maximum filter rules that apply to IntellaFlex blades. These are as follows:

- E18-6: 1536 maximum available per port. In general, 12,288 per blade.
- E24-1: 4096 maximum available per port. In general, 16,384 per blade.
- E24-2: 4096 maximum available per port. In general, 16,384 per blade.
- E36-1: 4096 maximum available per port. In general, 16,384 per blade.

To Maximize Filtering Efficiency

- Use a pass filter instead of a drop filter
- Use ingress instead of egress filtering

One of the most efficient practices to filter a range of addresses is to use bitmasks. The bitmask matches only a preset number of bits in the address (see below).

Minimize Filter Rules

To minimize the number of filter rules and avoid maximum filter rule limits:

- Use bitmask instead of range, when filtering on these parameters:
  - IPv4/IPv6 Source and Destination IP Addresses
  - Source and Destination MAC Addresses
- Avoid the AND operator wherever possible (AND has a multiplicative effect)

Types of Filtering Questions

Should I create a 1000-line filter with multiple "and" and "or" statements?

- Each filter statement breaks into a single or multiple rules in the filtering engine
  - The number of rules generated by the filtering engine depends on the equality operator (=, >, < etc.)
    - Use "=" wherever possible
  - The number of rules generated by the filtering engine depends on the Boolean logic parameter you use (Boolean parameters include and, or, not)
    - Avoid using "and" when you do not also use "="
  - The number of rules generated by the filtering engine, when you do not use the "=" operator, depends on the filtering parameter bit width
    - IP protocol uses 8 bits, IPv4 uses 32 bits, IPv6 uses 128 bits, TCP/UDP port uses 16 bits
  - Use these general tips for filtering on different parameters with the AND operator:
    - tcp.srcport > 20 AND ip.proto < 58 (will generate 16 × 8 rules because different parameters are used with the AND operator)

"The number of rules generated by the filtering engine depends on the Boolean logic parameter you use"

- There is a special case when using the same parameter for a range within the AND statement:
  - tcp.scrport > 20 AND tcp.scrport < 700 (will generate 16 rules instead of 16 × 16 rules because the same parameter is used with the AND operator)
  - This range can use the operators "<", "<=", ">" or ">=" with same effect.
- See the diagrams in the APCON *WEBX User Manual*, Appendix C, for specific information on additional protocols.

Given limitations on memory, how can I avoid Ternary Content Addressable Memory (TCAM) overflow errors?

- Distribute your filtering ports farther apart
  - On an E18-6 blade, for example, distribute your ingress ports into 2 groups. Place the first group on ports 1–9, and the second group on ports 10–18.
  - On an E18-6, for example, distribute your egress ports into 2 groups. Place the first group on ports 1–9, and the second group on ports 10–18.
  - Make sure your filter is efficiently written. See the examples provided above and in the Filtering chapter of the APCON *WEBX User Manual*.

When does egress filtering take place?

- Egress filtering takes place after packet aggregation has occurred and immediately before the each packet exits via the physical egress port on the switch.

Example Filter Rule Generation

or	(1 rule)
and	(multiplication of rules)
ip.src and ip.dst	(1 rule)
tcp.srcport < 1000 and tcp.dstport = 100	(16 × 1 rules)
tcp.srcport < 1000 and tcp.dstport < 100	(16 × 16 rules)
tcp.srcport = 1000	(1 rule)
tcp.srcport != 1000	(16 rules)
tcp.srcport = 1000 or tcp.srcport = 100	(2 rules)
tcp.srcport != 1000 and tcp.srcport != 100	(16 × 16 rules)
tcp.srcport > 20 and tcp.srcport < 1400	(16 instead of 16 × 16; explained above)
pass (ip.src != 10.5.168.19 && ip.dst != 10.5.168.123)	(32 × 32 rules)
drop (ip.src = 10.5.168.19 || ip.dst = 10.5.168.123)	(2 rules)

About APCON

APCON develops innovative, scalable technology solutions to enhance network monitoring, support IT traffic analysis, and streamline IT network management and security. APCON is the industry leader for state-of-the-art IT data aggregation, filtering, and network switching products, as well as leading-edge management-software support. Organizations in over 50 countries depend on APCON network infrastructure solutions. Customers include Global Fortune 500 companies, banks and financial services institutions, telecommunication service providers, government and military, and computer equipment manufacturers.

APCON Professional Services
Juan Garza
Director of Professional Services

971–224–2739
juan.garza@apcon.com

Example Filter Syntax

The following examples efficiently perform very common filtering tasks.

- **Monitor 4 servers (8 rules; 9 WebX filter statements):**

  pass (ip.src = 159.10.23.7 || ip.dst = 159.10.23.7 || ip.src = 159.10.23.57 || ip.dst = 159.10.23.57 || ip.src = 159.10.23.63 || ip.dst = 159.10.23.63 || ip.src = 159.10.23.75 || ip.dst = 159.10.23.75)

- **Monitor a whole subnet of max. 510 IP hosts (2 rules; 3 WebX filter statements):**

  pass (ip.src = 159.53.68.0/23 || ip.dst = 159.53.68.0/23)

- **Monitor all web traffic for a whole subnet of max. 510 IP hosts (12 rules; 10 WebX filter statements):**

  pass ((ip.src = 159.53.68.0/23 || ip.dst = 159.53.68.0/23) && (tcp.srcport = 80 || tcp.dstport = 80 || udp.srcport = 80 || udp.dstport = 80 || tcp.srcport = 443 || tcp.dstport = 443))

- **Exclude traffic for 2 servers and specific ports (16 rules; 11 WebX filter statements):**

  drop ((ip.src = 192.168.0.3 || ip.src = 192.168.0.5 || ip.dst = 192.168.0.3 || ip.dst = 192.168.0.5) && (tcp.srcport = 1150 || tcp.srcport = 1275 || tcp.dstport = 1150 || tcp.dstport = 1275))

Conclusion

By applying these simple and effective practices, you can make your filters more efficient, maximize system resources and improve network performance. To create custom filters, consider the examples shown above as a set of syntax examples that can be copied and edited to produce effective and comprehensive filters. For additional consultation or questions, please contact your APCON area representative or the Professional Services staff at support@apcon.com.