

Hoe om te gaan met microbursts in ASR 920, verschillen tussen bytes en percentage met wachtrij-limiet

Inhoud

[Inleiding](#)

[Voorwaarden](#)

[Vereisten](#)

[Gebruikte componenten](#)

[Achtergrondinformatie](#)

[Wachtrij instellen](#)

[Verschillen met RSP3 en NCS 520](#)

[Queue-Limit \(QL\) Lab-test, verschillen tussen bytes en percentage](#)

[Testbereik en -parameters](#)

[Testprocedure](#)

[Voorbeeld](#)

[Testresultaten van lab](#)

[QL procent tegenover bytes met 64 bytes](#)

[QL procent versus bytes met 200 bytes aan pakketten](#)

[QL procent tegenover bytes met 300 bytes](#)

[QL procent tegenover bytes met 518 bytes](#)

[QL procent tegenover bytes met 800 bytes](#)

[QL procent versus bytes met 1024 bytes aan pakketten](#)

[QL procent versus bytes met 1400 bytes aan pakketten](#)

[Relatie tussen drop en pakketgrootte](#)

[Verschil tussen equivalente bytes en procentuele waarden](#)

[Conclusies](#)

[Hoe de procentuele waarden in reële verkeersscenario's te benaderen](#)

[Hoe het gebruik van de microburstbuffer te controleren](#)

Inleiding

Dit document beschrijft hoe microbursts in ASR 920-routers moeten worden verwerkt, die vaak de oorzaak zijn van pakketdalingen in interfaceuitvoeringen. In detail wordt het verschil getoond tussen het gebruik van bytes en procenten met de wachtrij-limiet opdracht.

Voorwaarden

Vereisten

Cisco raadt kennis van de volgende onderwerpen aan:

Gebruikte componenten

De informatie in dit document is gebaseerd op een ASR 920 router die softwareversie Cisco IOS-XE 16.9.6 uitvoert.

IXIA wordt gebruikt als verkeersgenerator van de laboratoriumtest.

De informatie in dit document is gebaseerd op de apparaten in een specifieke laboratoriumomgeving. Alle apparaten die in dit document worden beschreven, hadden een opgeschoonde (standaard)configuratie. Als uw netwerk live is, moet u zorgen dat u de potentiële impact van elke opdracht begrijpt.

Achtergrondinformatie

Microbursts worden aangeduid als kleine spikes in netwerkverkeer. In Service Provider-toegangsnetwerken, bijvoorbeeld, komt dit vaak voor in snelheids-mismatch-scenario's waarbij een verkeersstroom de router ingaat via een snelle interface, zoals 10 Gigabit Ethernet (GE), en zich uitdrukt via een interface met lage snelheid, zoals 1 GE.

Het meest voorkomende probleem dat microbutsts in ASR 920 routers veroorzaken is de daling van het interfacepakket in outputrichting. Dit gebeurt wanneer er uitbarstingen op de inkomende interface zijn die hoger zijn dan het tarief van de uitgaande interface, voor een zeer korte periode (orde van milliseconden). Gedurende deze tijd moeten de pakketten worden gebufferd.

Op ASR 920 / RSP2-platforms is de standaardbuffer die voor wachtrijen op elke 1 GE-interface wordt toegewezen 48 KB, terwijl voor de wachtrijen op elke 10 GE-interface 120 KB is.

Bovendien is er een gedeelde buffer beschikbaar van 11,75 MB. Wanneer de standaardinterfacebuffer niet genoeg is om een uitbarsting van pakketten aan te passen, kan de gedeelde buffer worden gebruikt.

Om het gebruik van de gedeelde buffer voor een interface toe te laten, is het nodig om een QoS-beleid onder die interface te configureren, die de wachtrij-limiet parameter definieert.

Wachtrij instellen

Dit is een configuratievoorbeeld van wachtrij-limiet voor interface GE 0/0/1:

```
class-map match-all DUMMY
match qos-group 99 <--- it can be any unused group
```

```
policy-map QUEUE-LIMIT
class DUMMY
class class-default
queue-limit percent 5
```

```
interface GigabitEthernet 0/0/1
service-policy output QUEUE-LIMIT
```

Dit beleid komt niet overeen met enig verkeer en heeft dus ook geen effect. Al dit beleid doet is de

wachtrijbuffer van interface GE 0/0/1 te verhogen.

De opdracht **match qos-group 99** is nodig omdat het niet wordt ondersteund om wachtrij-limiet te configureren in een niet-bladklasse, zodat u een fictieve klasse-kaart als ouderklasse moet gebruiken om de wachtrij-limiet opdracht te configureren onder de bladklasse-standaard:

```
ASR-920-1(config)#policy-map QUEUE-LIMIT
ASR-920-1(config-pmap)#class class-default
ASR-920-1(config-pmap-c)# queue-limit percent 5
QoS: queue-limit command not supported in non-leaf classes
queue-limit: platform params check fail
```

De wachtrij-limiet kan op verschillende manieren worden ingesteld:

```
(config-pmap-c)#queue-limit ?
<1-2097152> in bytes, <1-1677721> in us, <1-8192000> in packets by default
percent % of threshol
```

De parameter en de waarde moeten in overeenstemming met uw netwerkvereisten worden geselecteerd.

Om die parameters te begrijpen en hoe een verandering in de waarden het buffergebruik beïnvloedt, kunt u naar de laboratoriumtest verwijzen die in de volgende sectie wordt voorgesteld.

Opmerking: het wordt niet ondersteund om een QoS-beleid te configureren onder poortkanaal-interfaces, alleen onder het fysieke interfacegedeelte van het poortkanaal

```
ASR-920-1(config)#interface port-channel 2
ASR-920-1(config-if)#service-policy output qos-tac
QoS: Configuration failed. Policy-map with Queueing actions not supported on EC main-
interface/EFP
QoS: Configuration errors for policymap qos-tac
```

Verschillen met RSP3 en NCS 520

[Cisco RSP3-module - QoS-mogelijkheden:](#)

- RSP3-module heeft 4 GB externe pakketbuffers per NPU
- RSP3-module ondersteunt 48000 wachtrijen
- Standaard ondersteunt RSP3-module tot 1 MB wachtrij-limiet per wachtrij
- Queue limit percentage wordt beschouwd als uit 1 GB van de totale buffers

Voor routers met RSP3-supervisors en NCS 520 is het aantal qos-groepen dat kan worden geconfigureerd beperkt tot 0-7:

```
ASR-903-1(config)#class-map match-all qos-tac
ASR-903-1(config-cmap)#match qos-group ?
<0-7> Qos Group value
```

In de NCS 520 is er standaard een wachtrijbuffer van 2 MB gedeeld tussen alle interfaces, een externe 2 GB-buffer is toegankelijk wanneer een policy-map met wachtrijlimiet is geconfigureerd. Er is ook een verschil in de bytes en de gebruiksparementen voor de wachtrij-limiet:

```
ASR-520-1(config-pmap-c)# queue-limit ?
<1-8192000> in bytes, <1-40000> in us, <1-8192000> in packets by default
percent % of threshold
```

Queue-Limit (QL) Lab-test, verschillen tussen bytes en percentage

Testbereik en -parameters

Zoals eerder waargenomen is de maximale configureerbare waarde voor wachtrij-limiet in bytes 2097152, wat ongeveer 18% is van de gedeelde buffer op ASR 920-platforms (11,75 MB ~ 45898 * 256 bytes).

Als je wachtrij-limiet instelt in procent, kun je tot 100% gaan. Om de percentages en bytes met equivalente waarden te vergelijken, gebruikt de test de bytewaarden van 117498 tot 2097152 bytes en de wachtrijlimietwaarden van 1% tot 18%:

```
queue-limit percent 1 <=> queue-limit 117498 bytes
queue-limit percent 2 <=> queue-limit 234996 bytes
queue-limit percent 3 <=> queue-limit 352494 bytes
queue-limit percent 4 <=> queue-limit 469992 bytes
queue-limit percent 5 <=> queue-limit 587490 bytes
queue-limit percent 6 <=> queue-limit 704988 bytes
queue-limit percent 7 <=> queue-limit 822486 bytes
queue-limit percent 8 <=> queue-limit 939984 bytes
queue-limit percent 9 <=> queue-limit 1057482 bytes
queue-limit percent 10 <=> queue-limit 1174980 bytes
queue-limit percent 11 <=> queue-limit 1292478 bytes
queue-limit percent 12 <=> queue-limit 1409976 bytes
queue-limit percent 13 <=> queue-limit 1527474 bytes
queue-limit percent 14 <=> queue-limit 1644972 bytes
queue-limit percent 15 <=> queue-limit 1762470 bytes
queue-limit percent 16 <=> queue-limit 1879968 bytes
queue-limit percent 17 <=> queue-limit 1997466 bytes
queue-limit percent 18 <=> queue-limit 2097152 bytes
```

36 beleidskaarten zijn geconfigureerd: 18 met wachtrij-grenswaarden die variëren van 1% tot 18% en de andere 18 met wachtrij-grenswaarden die variëren van 117498 tot 2097152 bytes.

```
policy-map QUEUE-LIMIT-PERCENT-X
class DUMMY
class class-default
  queue-limit percent X
```

```
policy-map QUEUE-LIMIT-BYTES-X
class DUMMY
class class-default
  queue-limit Y bytes
```

=> X values range from 1 to 18

=> Y values range from 117498 to 2097152

Elk beleid wordt getest tegen hetzelfde microburstverkeer, dat met IXIA wordt gegenereerd. Dit verkeer komt aan op een 10 GE-poort van de ASR 920 en sluit af op een 1 GE-poort van dezelfde router.

De uitbarstingen bestaan uit 20000 pakketten bij 4 Gbps met intervallen van 5 seconden. Dit is de duur van de burst bij een specifieke pakketgrootte:

1280000 bytes at 64 packet size, Burst duration: 0.00256 second
4000000 bytes at 200 packet size, Burst duration: 0.008 second
6000000 bytes at 300 packet size, Burst duration: 0.012 second
10360000 bytes at 518 packet size, Burst duration: 0.02072 second
16000000 bytes at 800 packet size, Burst duration: 0.032 second
20480000 bytes at 1024 packet size, Burst duration: 0.04096 second
28000000 bytes at 1400 packet size, Burst duration: 0.056 second

Testprocedure

Stap 1. Pas de policy-map QUEUE-limit-BYTES-X (X=1) met bytes Y=117498 toe onder de uitgaande interface.

Stap 2. Start microburst traffic 1 minuut.

Stap 3. Meet het totale aantal pakketten in de uitvoer en hoeveel pakketten zijn gevallen.

Stap 4. Bereken de verhouding tussen gedropte pakketten en totale uitvoerpakketten.

Stap 5. Herhaal één keer van stap 1, deze keer gebruiken beleid-kaart WACHTRIJ-LIMIET-PERCENT-X met rij-limiet percent X, waar X=1.

Stap 6. Herhaal uit stap 1 met X=X+1 voor de naam van de beleidskaart en de procentuele waarde, de bytes Y=Y+117498. Herhaal dit tot X=18 en Y=2097152.

Voorbeeld

Meting met QUEUE-limit-BYTES-1 policy-map:

```
ASR-920-1#show int Gi0/0/1
GigabitEthernet0/0/1 is up, line protocol is up
  Hardware is 24xGE-4x10GE-FIXED-S, address is 70df.2f2f.ed01 (bia 70df.2f2f.ed01)
  Internet address is 10.12.10.47/31
  MTU 8900 bytes, BW 1000000 Kbit/sec, DLY 10 usec,
    reliability 255/255, txload 4/255, rxload 4/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full Duplex, 1000Mbps, link type is auto, media type is T
  output flow-control is unsupported, input flow-control is on
  Carrier delay is 0 msec
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:01, output 00:00:02, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/375/0/0 (size/max/drops/flushes); Total output drops: 684263427
  Queueing strategy: Class-based queueing
  Output queue: 0/40 (size/max)
  30 second input rate 19475000 bits/sec, 19533 packets/sec
  30 second output rate 19157000 bits/sec, 13356 packets/sec
    5064106237 packets input, 4333296255278 bytes, 0 no buffer
  Received 29 broadcasts (0 IP multicasts)
    0 runs, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
    0 watchdog, 726180 multicast, 0 pause input
  7829367523 packets output, 4217074973677 bytes, 0 underruns
    0 output errors, 0 collisions, 3 interface resets
    0 unknown protocol drops
    0 babbles, 0 late collision, 0 deferred
    0 lost carrier, 0 no carrier, 0 pause output
```

0 output buffer failures, 0 output buffers swapped out

ASR-920-1#**show policy-map int Gi0/0/1 output**
GigabitEthernet0/0/1

Service-policy output: QUEUE-LIMIT-BYTES-1

Class-map: DUMMY (match-all)
0 packets, 0 bytes
30 second offered rate 0000 bps
Match: qos-group 99

Class-map: class-default (match-any)
1044078 packets, 73085460 bytes
30 second offered rate 9759000 bps, drop rate 0000 bps
Match: any

queue limit 117498 bytes
(queue depth/total drops/no-buffer drops) 0/0/0
(pkts output/bytes output) 0/0

Meting na 1 minuut microburst-verkeer:

ASR-920-1#**show int Gi0/0/1**
GigabitEthernet0/0/1 is up, line protocol is up
Hardware is 24xGE-4x10GE-FIXED-S, address is 70df.2f2f.ed01 (bia 70df.2f2f.ed01)
Internet address is 10.12.10.47/31
MTU 8900 bytes, BW 1000000 Kbit/sec, DLY 10 usec,
reliability 255/255, txload 2/255, rxload 3/255
Encapsulation ARPA, loopback not set
Keepalive set (10 sec)
Full Duplex, 1000Mbps, link type is auto, media type is T
output flow-control is unsupported, input flow-control is on
Carrier delay is 0 msec
ARP type: ARPA, ARP Timeout 04:00:00
Last input 00:00:01, output 00:00:01, output hang never
Last clearing of "show interface" counters never
Input queue: 0/375/0/0 (size/max/drops/flushes); Total output drops: 684561562
Queueing strategy: Class-based queueing
Output queue: 0/40 (size/max)
30 second input rate 13981000 bits/sec, 19643 packets/sec
30 second output rate 11256000 bits/sec, 12784 packets/sec
5064715137 packets input, 4333338878716 bytes, 0 no buffer
Received 29 broadcasts (0 IP multicasts)
0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
0 watchdog, 726190 multicast, 0 pause input
7829753878 packets output, 4217102018968 bytes, 0 underruns
0 output errors, 0 collisions, 3 interface resets
0 unknown protocol drops
0 babbles, 0 late collision, 0 deferred
0 lost carrier, 0 no carrier, 0 pause output
0 output buffer failures, 0 output buffers swapped out

ASR-920-1#**show policy-map int Gi0/0/1 output**
GigabitEthernet0/0/1

Service-policy output: QUEUE-LIMIT-BYTES-1

```
Class-map: DUMMY (match-all)
  0 packets, 0 bytes
  30 second offered rate 0000 bps
  Match: qos-group 99
```

```
Class-map: class-default (match-any)
  1847215 packets, 129305050 bytes
  30 second offered rate 10804000 bps, drop rate 0000 bps
  Match: any
```

```
queue limit 117498 bytes
(queue depth/total drops/no-buffer drops) 0/387570/0
(pkts output/bytes output) 656508/45955560
```

Packet druppels delta: $684561562 - 684263427 = 298135$

Totale output van pakketten: $7829753878 - 7829367523 = 386355$

Verhouding tussen pakketdalingen en pakketten: $298135 / 386355 = 77\%$

Testresultaten van lab

Zoals toegelicht worden 36 beleidskaarten getest: 18 geconfigureerd met wachtrij-limietwaarden die variëren van 1% tot 18% worden getest tegen de andere 18 beleidsgebieden geconfigureerd met wachtrij-limietwaarden die variëren van 117498 tot 2097152 bytes. Elke policy-map wordt getest tegen hetzelfde microburst verkeer, gegenereerd met IXIA.

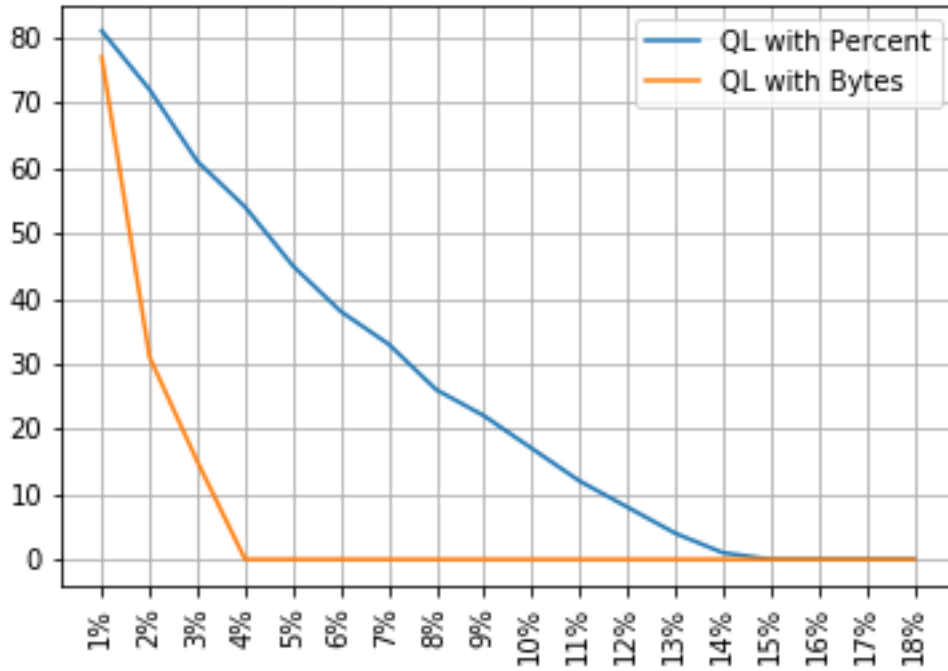
In deze paragraaf worden de resultaten van deze zeven maal herhaald om de resultaten met verschillende pakketgrootten in bytes te controleren: 64, 200, 300, 518, 800, 1024, 1400.

Om het lezen te vergemakkelijken worden de resultaten voor elke pakketgrootte in een tabel weergegeven en vervolgens geïllustreerd.

QL procent tegenover bytes met 64 bytes

Policy-map		Drop rate with PC	Drop rate with Bytes
QL Bytes/Percent	1	81%	77%
QL Bytes/Percent	2	72%	31%
QL Bytes/Percent	3	61%	15%
QL Bytes/Percent	4	54%	0%
QL Bytes/Percent	5	45%	0%
QL Bytes/Percent	6	38%	0%
QL Bytes/Percent	7	33%	0%
QL Bytes/Percent	8	26%	0%
QL Bytes/Percent	9	22%	0%
QL Bytes/Percent	10	17%	0%
QL Bytes/Percent	11	12%	0%
QL Bytes/Percent	12	8%	0%
QL Bytes/Percent	13	4%	0%
QL Bytes/Percent	14	1%	0%
QL Bytes/Percent	15	0%	0%
QL Bytes/Percent	16	0%	0%
QL Bytes/Percent	17	0%	0%
QL Bytes/Percent	18	0%	0%

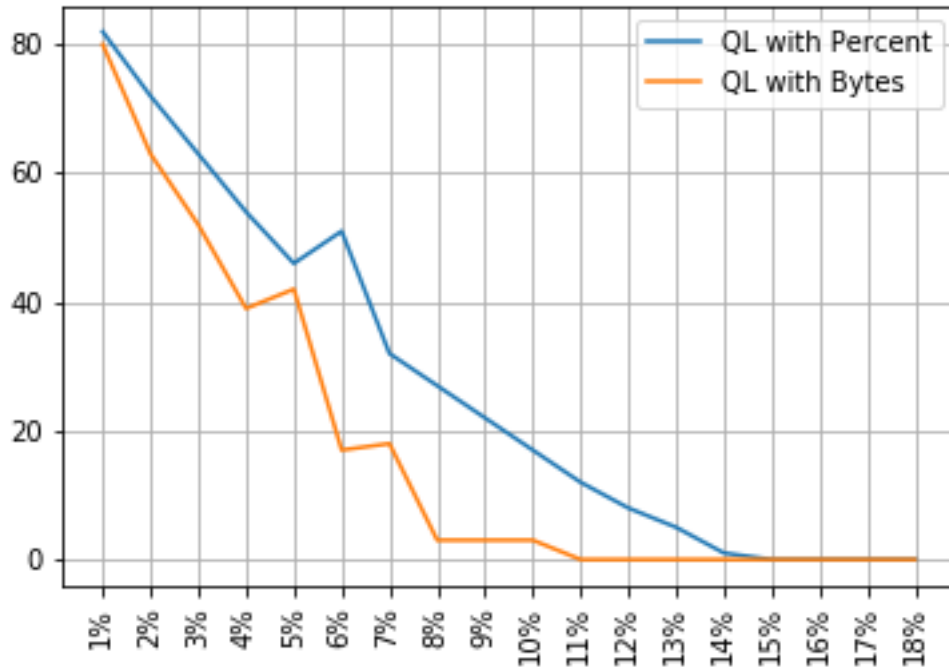
QL Percent vs bytes with 64 bytes packets



QL procent versus bytes met 200 bytes aan pakketten

Policy-map		Drop rate with PC	Drop rate with Bytes
QL Bytes/Percent 1		82%	80%
QL Bytes/Percent 2		72%	63%
QL Bytes/Percent 3		63%	52%
QL Bytes/Percent 4		54%	39%
QL Bytes/Percent 5		46%	42%
QL Bytes/Percent 6		51%	17%
QL Bytes/Percent 7		32%	18%
QL Bytes/Percent 8		27%	3%
QL Bytes/Percent 9		22%	3%
QL Bytes/Percent 10		17%	3%
QL Bytes/Percent 11		12%	0%
QL Bytes/Percent 12		8%	0%
QL Bytes/Percent 13		5%	0%
QL Bytes/Percent 14		1%	0%
QL Bytes/Percent 15		0%	0%
QL Bytes/Percent 16		0%	0%
QL Bytes/Percent 17		0%	0%
QL Bytes/Percent 18		0%	0%

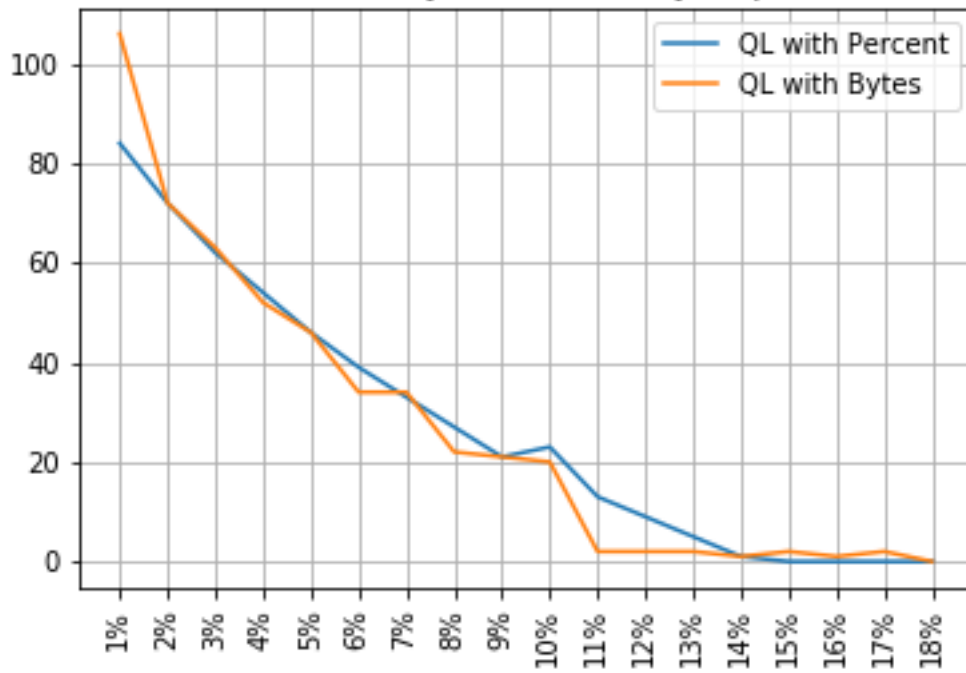
QL Percent vs bytes with 200 bytes packets



QL procent tegenover bytes met 300 bytes

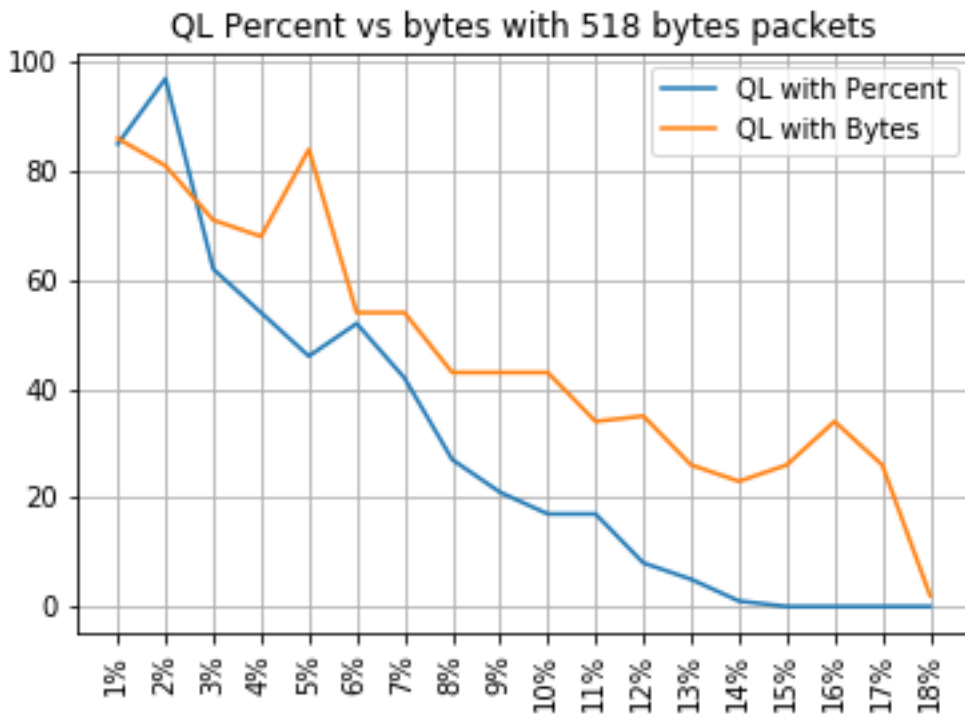
Policy-map	Drop rate with PC	Drop rate with Bytes
QL Bytes/Percent 1	84%	106%
QL Bytes/Percent 2	72%	72%
QL Bytes/Percent 3	62%	63%
QL Bytes/Percent 4	54%	52%
QL Bytes/Percent 5	46%	46%
QL Bytes/Percent 6	39%	34%
QL Bytes/Percent 7	33%	34%
QL Bytes/Percent 8	27%	22%
QL Bytes/Percent 9	21%	21%
QL Bytes/Percent 10	23%	20%
QL Bytes/Percent 11	13%	2%
QL Bytes/Percent 12	9%	2%
QL Bytes/Percent 13	5%	2%
QL Bytes/Percent 14	1%	1%
QL Bytes/Percent 15	0%	2%
QL Bytes/Percent 16	0%	1%
QL Bytes/Percent 17	0%	2%
QL Bytes/Percent 18	0%	0%

QL Percent vs bytes with 300 bytes packets



QL procent tegenover bytes met 518 bytes

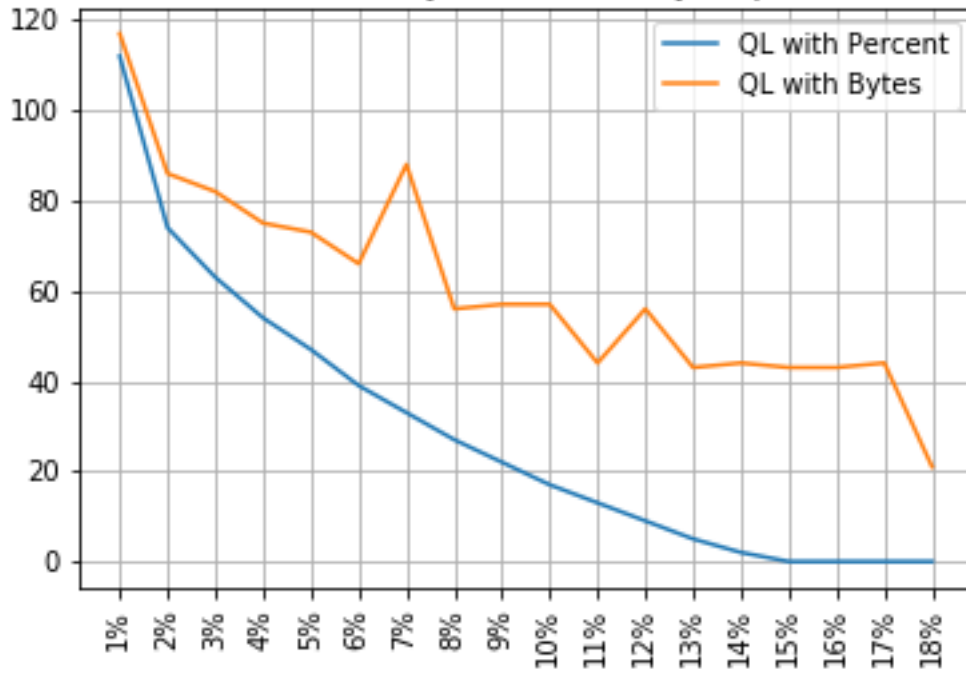
Policy-map	Drop rate with PC	Drop rate with Bytes
QL Bytes/Percent 1	85%	86%
QL Bytes/Percent 2	97%	81%
QL Bytes/Percent 3	62%	71%
QL Bytes/Percent 4	54%	68%
QL Bytes/Percent 5	46%	84%
QL Bytes/Percent 6	52%	54%
QL Bytes/Percent 7	42%	54%
QL Bytes/Percent 8	27%	43%
QL Bytes/Percent 9	21%	43%
QL Bytes/Percent 10	17%	43%
QL Bytes/Percent 11	17%	34%
QL Bytes/Percent 12	8%	35%
QL Bytes/Percent 13	5%	26%
QL Bytes/Percent 14	1%	23%
QL Bytes/Percent 15	0%	26%
QL Bytes/Percent 16	0%	34%
QL Bytes/Percent 17	0%	26%
QL Bytes/Percent 18	0%	2%



QL procent tegenover bytes met 800 bytes

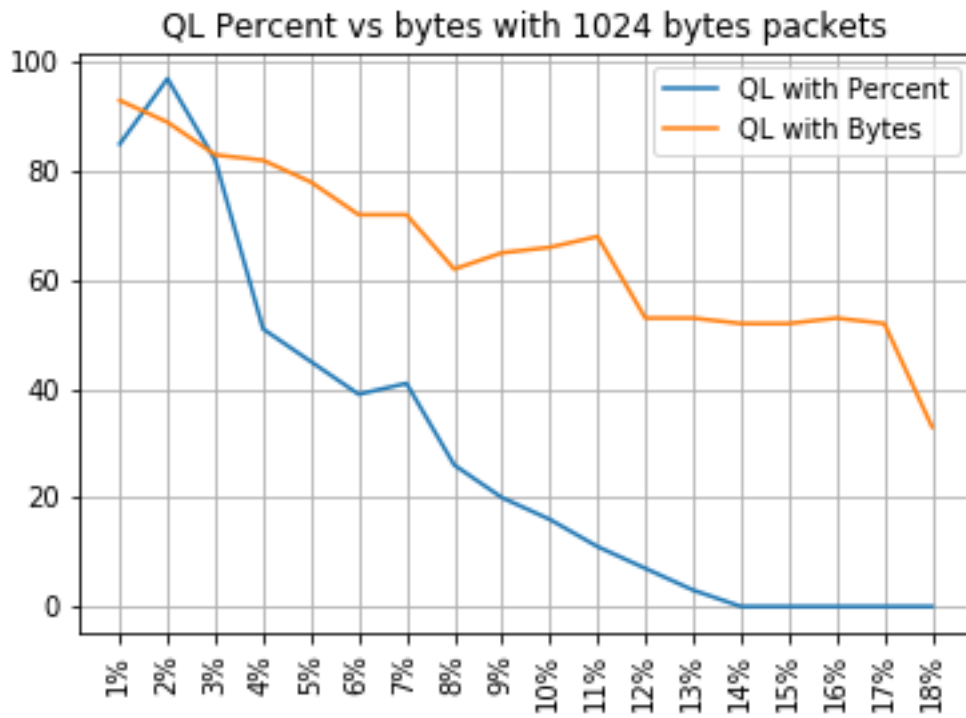
Policy-map	Drop rate with PC	Drop rate with Bytes
QL Bytes/Percent 1	112%	117%
QL Bytes/Percent 2	74%	86%
QL Bytes/Percent 3	63%	82%
QL Bytes/Percent 4	54%	75%
QL Bytes/Percent 5	47%	73%
QL Bytes/Percent 6	39%	66%
QL Bytes/Percent 7	33%	88%
QL Bytes/Percent 8	27%	56%
QL Bytes/Percent 9	22%	57%
QL Bytes/Percent 10	17%	57%
QL Bytes/Percent 11	13%	44%
QL Bytes/Percent 12	9%	56%
QL Bytes/Percent 13	5%	43%
QL Bytes/Percent 14	2%	44%
QL Bytes/Percent 15	0%	43%
QL Bytes/Percent 16	0%	43%
QL Bytes/Percent 17	0%	44%
QL Bytes/Percent 18	0%	21%

QL Percent vs bytes with 800 bytes packets



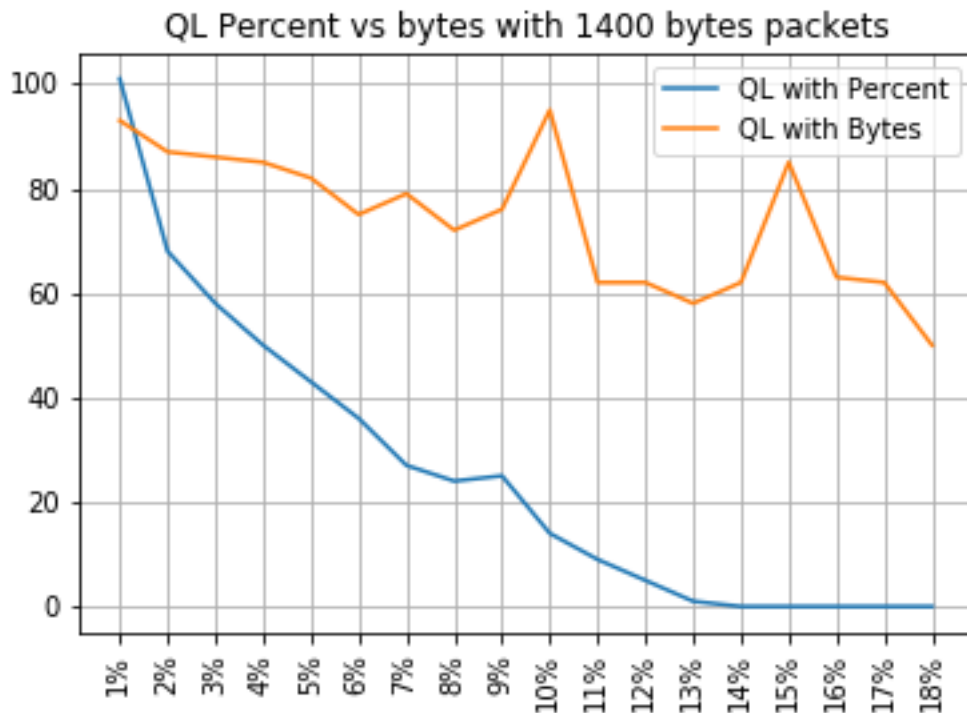
QL procent versus bytes met 1024 bytes aan pakketten

Policy-map	Drop rate with PC	Drop rate with Bytes
QL Bytes/Percent 1	85%	93%
QL Bytes/Percent 2	97%	89%
QL Bytes/Percent 3	82%	83%
QL Bytes/Percent 4	51%	82%
QL Bytes/Percent 5	45%	78%
QL Bytes/Percent 6	39%	72%
QL Bytes/Percent 7	41%	72%
QL Bytes/Percent 8	26%	62%
QL Bytes/Percent 9	20%	65%
QL Bytes/Percent 10	16%	66%
QL Bytes/Percent 11	11%	68%
QL Bytes/Percent 12	7%	53%
QL Bytes/Percent 13	3%	53%
QL Bytes/Percent 14	0%	52%
QL Bytes/Percent 15	0%	52%
QL Bytes/Percent 16	0%	53%
QL Bytes/Percent 17	0%	52%
QL Bytes/Percent 18	0%	33%



QL procent versus bytes met 1400 bytes aan pakketten

Policy-map	Drop rate with PC	Drop rate with Bytes
QL Bytes/Percent 1	101%	93%
QL Bytes/Percent 2	68%	87%
QL Bytes/Percent 3	58%	86%
QL Bytes/Percent 4	50%	85%
QL Bytes/Percent 5	43%	82%
QL Bytes/Percent 6	36%	75%
QL Bytes/Percent 7	27%	79%
QL Bytes/Percent 8	24%	72%
QL Bytes/Percent 9	25%	76%
QL Bytes/Percent 10	14%	95%
QL Bytes/Percent 11	9%	62%
QL Bytes/Percent 12	5%	62%
QL Bytes/Percent 13	1%	58%
QL Bytes/Percent 14	0%	62%
QL Bytes/Percent 15	0%	85%
QL Bytes/Percent 16	0%	63%
QL Bytes/Percent 17	0%	62%
QL Bytes/Percent 18	0%	50%



Relatie tussen drop en pakketgrootte

Zoals vermeld heeft de ASR 920 een interne pakketbuffer van 11,75 MB, die is verdeeld in 45898 Qnodes van 256 bytes elk.

- Voor een pakket met een grootte < 256 bytes wordt precies 1 Qnode gebruikt
- Voor een pakket met een grootte van 1024 bytes worden 4 Qnodes gebruikt
- Voor een pakket met 257 bytes worden 2 Qnodes gebruikt en gaan de 255 bytes die niet worden gebruikt, verloren

Daarom kun je een kleinere hoeveelheid grote pakketten opslaan dan wat je met kleine pakketten kan opslaan. Het verband tussen dalingstarief en pakketgrootte bij gelijkwaardige rij-grens grootte wordt verwacht.

Verschil tussen equivalente bytes en procentuele waarden

Zoals uitgelegd zijn er 45898 Qnodes in de 11,75 MB gedeelde buffer, afgerond naar 45900 voor eenvoudige berekening.

De berekening van het aantal wachtrijen-limieten berekent niet het percentage van de 11,75 MB, maar dat van de 45900 Qnodes. Dus, wachtrij-limiet procent 10 betekent 10% van 45900 wat 4590 Qnodes geeft.

Bovendien wordt het percentage toegewezen Qnodes beschouwd als het aantal pakketten dat in de wachtrij kan worden opgeslagen, onafhankelijk van de grootte ervan. Terug naar het vorige voorbeeld, betekent dit dat:

wachtrij-limiet percentage 10 = 4590 Qnodes = 4590 pakketten.

Aangezien deze berekening onafhankelijk is van de pakketgrootte, wordt voor pakketten met een grootte van 256 bytes of minder slechts één Qnode daadwerkelijk gebruikt en blijft de equivalentie tussen Qnodes en pakketten behouden:

wachtrij-limiet percentage 10 = 4590 Qnodes = 4590 pakketten van 256 bytes = 4590*256 bytes = 1,175 MB = 10% van de buffer

Met grotere pakketten echter, wordt een genereuzer deel van de buffer toegewezen. Dit is bijvoorbeeld de berekening voor 1024 bytes-pakketten, waarbij elk pakket 4 Qnodes verbruikt:

wachtrij-limiet percentage 10 = 4590 Qnodes = 4590 pakketten van 1024 bytes = 4590*4*256 bytes = 4,7 MB = 40% van de buffer

Voorzichtig: het wordt niet aanbevolen om hoge waarden van wachtrij-limiet procent te configureren.

Als u hoge waarden van wachtrij-limiet procent vormt, kan één interface tijdelijk alle gedeelde buffer van 11,75 MB innemen.

Conclusies

- U kunt duidelijk zien dat de efficiëntie van wachtrij-limiet bytes beter is met kleine pakketten - wachtrij-limiet bytes <x> werkt beter dan wachtrij-limiet <x> voor maximaal 300 bytes
- Bij een pakket van 300 bytes zijn wachtrij-limietbytes en wachtrij-limiet percent efficiëntie hetzelfde
- Met een pakketgrootte van meer dan 300 bytes is wachtrijgrenswaarde efficiënter. Aangezien internetverkeer gemiddeld 518 bytes telt, betekent dit dat real-life scenario's meer profiteren van wachtrijlimietpercentage, zoals gemeld door de klanten
- De efficiëntie van wachtrij-limiet percent lineair verbetert gerelateerd aan de pakketgrootte (hoe groter de pakketten de efficiëntere wachtrij-limiet percent is vs wachtrij-limiet bytes)
- Queue-limit percentage is geïmplementeerd om genereuzer te zijn in de toewijzing van bufferruimte voor pakketten van meer dan 256 MB

Hoe de procentuele waarden in reële verkeersscenario's te benaderen

In het geval waar u pakketten met grootte van 256 bytes en rij-limiet van 10% hebt, weet u reeds dat deze equivalentie geldig is:

`minimum queue-limit = 4590 Qnodes = 4590*256 bytes = 1.175 MB = 10% of the buffer`

Met 512 bytes pakketten is alleen het gebruik de dubbele, met 1024 bytes alleen pakketten is het vier keer zo veel, enzovoort.

Dit betekent dat de werkelijke wachtrij-limiet minimaal 10% van de buffer is en als je uitgaat van een maximale MTU van 1500 bytes, heb je 6 Qnodes nodig om één pakket op te slaan, wat een maximale wachtrij-limiet geeft van:

`maximum queue-limit = 4590*6 Qnodes = 4590*256*6 bytes = 7.05 MB = 60% of the buffer`

Op deze manier kunt u de onder- en bovengrenzen van het buffergebruik definiëren met wachtrij-limiet procent 10, dus meer in het algemeen is het gemiddelde max buffergebruik ongeveer:

```
ceil(avg_pkt_size/256)*((qlimit_percent/45900)*100)
```

Voorbeeld van een laboratoriumapparatuur:

```
GigabitEthernet0/0/1 is up, line protocol is up
Hardware is 24xGE-4x10GE-FIXED-S, address is 70df.2f2f.ed01 (bia 70df.2f2f.ed01)
Internet address is 10.12.10.47/31
MTU 8900 bytes, BW 1000000 Kbit/sec, DLY 10 usec,
reliability 255/255, txload 25/255, rxload 30/255
Encapsulation ARPA, loopback not set
Keepalive set (10 sec)
Full Duplex, 1000Mbps, link type is auto, media type is T
output flow-control is unsupported, input flow-control is on
Carrier delay is 0 msec
ARP type: ARPA, ARP Timeout 04:00:00
Last input 00:00:00, output 00:00:01, output hang never
Last clearing of "show interface" counters 00:11:43
Input queue: 0/375/0/0 (size/max/drops/flushes); Total output drops: 2036062
Queueing strategy: Class-based queueing
Output queue: 0/40 (size/max)
30 second input rate 118520000 bits/sec, 18902 packets/sec
30 second output rate 101646000 bits/sec, 16124 packets/sec
13185272 packets input, 10328798549 bytes, 0 no buffer
Received 0 broadcasts (0 IP multicasts)
0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
0 watchdog, 235 multicast, 0 pause input
11247114 packets output, 8870166880 bytes, 0 underruns <<< avg_pkt_size = 8870166880/11247114 =
788.66 bytes
0 output errors, 0 collisions, 0 interface resets
0 unknown protocol drops
0 babbles, 0 late collision, 0 deferred
0 lost carrier, 0 no carrier, 0 pause output
0 output buffer failures, 0 output buffers swapped out
```

Je kunt de avg_pkt_size berekenen als $8870166880/11247114 \sim 788$ bytes.

Met een wachtrij-limiet procent 10, hebt u een gemiddeld max buffergebruik van:

```
ceil(avg_pkt_size/256)*((45900/100)*qlimit_percent)
```

Calculation example with Python:

```
>>>import math
>>>math.ceil(788/256)*((45900/100)*10)
18360.0
```

=> 18360 Qnodes = 18360 * 256 bytes = 4.7 MB = 40% of the buffer

Hoe het gebruik van de microburstbuffer te controleren

In Cisco IOS-XE releases voorafgaand aan 16.9.3, werd de gedeelde buffer van de interfaces gebruikt voor zowel gegevens- als controlepakketten (zoals BFD, routingprotocollen, ARP, LDP, punt keepalives). Om het gebruik van de onmiddellijke buffer te verifiëren kunt u deze opdracht gebruiken:

```
ASR-920-1#request platform software sdcli "nile bm reg buffertablefreelistcount show 0 0 0"
```

Na 16.9.3 zijn enkele wijzigingen geïntroduceerd om het buffergebruik te verbeteren en het is opgesplitst in 2: 1024 items (256KB) zijn gereserveerd voor controleverkeer en de rest is

gereserveerd voor dataverkeer.

Het buffergebruik kan in dit geval met deze opdrachten worden bewaakt:

```
ASR-920-1#request platform software sdcli "nile bm reg
supervisorresourcereservedcounttableaccess sh 0 0 0"
reservedUsedCount = 48 (0x30)
reservedFreeCount = 976 (0x3d0)
```

```
ASR-920-1#request platform software sdcli "nile bm reg
supervisorresourcereservedcounttableaccess sh 0 2 0"
reservedUsedCount = 8114 (0x1fb2)
reservedFreeCount = 37784 (0x9398)
```

Merk op dat, gezien het feit dat de buffer microbursts behandelt, u de opdracht vele malen moet herhalen om de gereserveerdeUsedCount waarde anders van 0 te zien.

Het buffergebruik kan simpelweg worden berekend met $\text{gereserveerdeUsedCount} / \text{gereserveerdeFreeCount}$, bijvoorbeeld $8114 / 37784 = 21,5\%$ gebruikt. Als de barst voorbij is, moet de buffer snel terugvallen naar 0 of dicht bij.

Van Cisco IOS-XE release 17.6.1 is het mogelijk om de hele buffer voor zowel gegevens- als controleverkeer te gebruiken (bij voorkeur als uw netwerk een hoog controleverkeer heeft) of de buffer in 2 te splitsen zoals eerder beschreven. De keuze wordt gemaakt met de configuratie van deze instructie (standaard uitgeschakeld):

```
ACDC-920-1(config)#platform qos-buffer enhance enable
ACDC-920-1(config)#no platform qos-buffer enhance enable
```

Van Cisco IOS-XE release 17.7.1 is het ook mogelijk de grootte te kiezen om toe te wijzen aan het controleverkeer:

```
ACDC-920-1(config)#platform qos-buffer enhance [1-4]
```

waarbij:

- 1 geeft een controlebuffer aan van 256 KB
- 2 geeft 500 KB aan
- 3 geeft 756 KB aan
- 4 geeft 1 MB aan

Over deze vertaling

Cisco heeft dit document vertaald via een combinatie van machine- en menselijke technologie om onze gebruikers wereldwijd ondersteuningscontent te bieden in hun eigen taal. Houd er rekening mee dat zelfs de beste machinevertaling niet net zo nauwkeurig is als die van een professionele vertaler. Cisco Systems, Inc. is niet aansprakelijk voor de nauwkeurigheid van deze vertalingen en raadt aan altijd het oorspronkelijke Engelstalige document ([link](#)) te raadplegen.