BGP

- IBGP
- EBGP
- Filtering, redistribution, summarization, synchronization, attributes and other advanced features

BGP Route Selection Criteria

- Prefer highest weight (local to router)
- Prefer highest local preference (global within AS)
- Prefer routes that the router originated
- Prefer shorter AS paths (only length is compared)
- Prefer lowest origin code (IGP < EGP < Incomplete)
- Prefer lowest MED
- Prefer external (EBGP) paths over internal (IBGP)
- For IBGP paths, prefer path through closest IGP neighbor
- For EBGP paths, prefer oldest (most stable) path
- Prefer paths from router with the lower BGP router-ID

Commonly Used Characters in Expressions

.	Any single character, including a space
*	Zero or more sequence of pattern
+	One or more sequence of pattern
?	Zero or one occurrence of pattern
^	Beginning of string
$	End of string
_	Match any delimiter (including beginning, end, space, tab, comma)
\	Remove special meaning of character that follows
[ ]	Match one character in a range
|	Logical OR

String Matching: Special Characters

\ To use the special characters as single-character patterns, remove the special meaning by preceding each character with a backslash (\).

How do you match AS 213 in the beginning of the string? `|(213 317) 1218 316 31|`

Answer: `^\(213_`

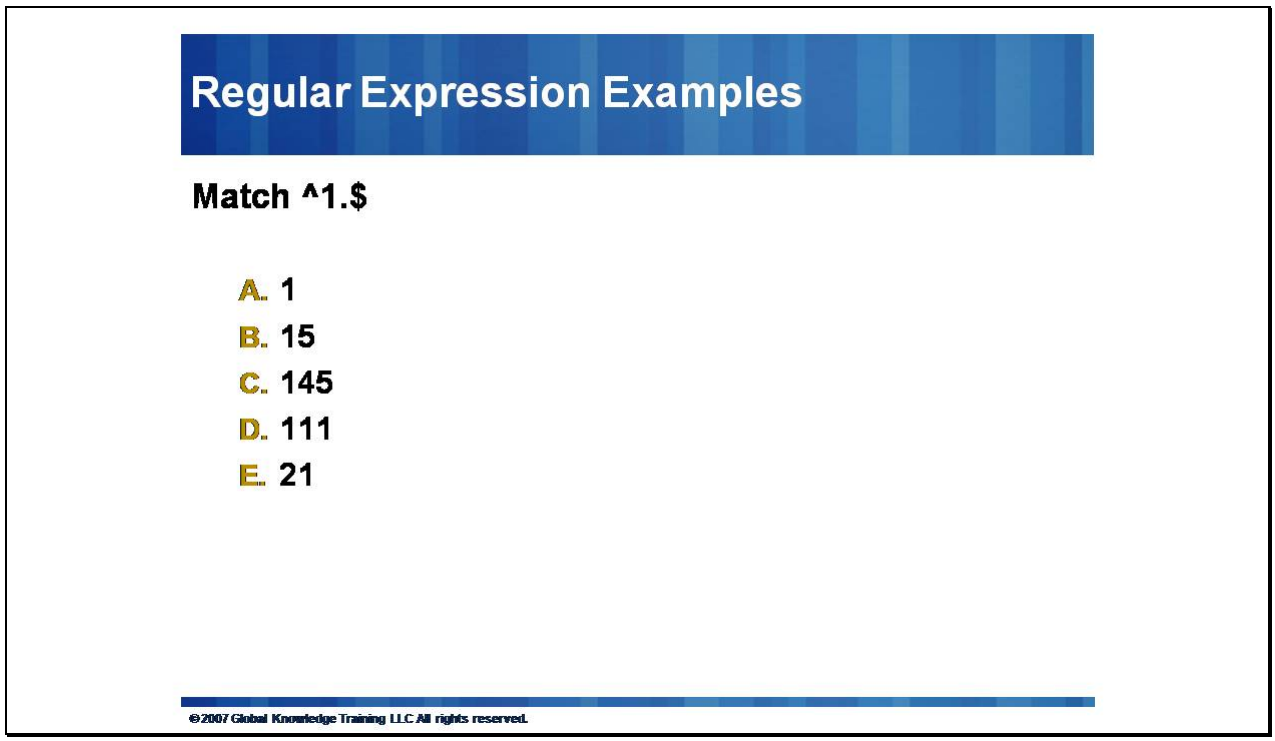
String Matching: Repeating Operators

*	Matches zero or more atoms
?	Matches zero or one atom
+	Matches one or more atoms

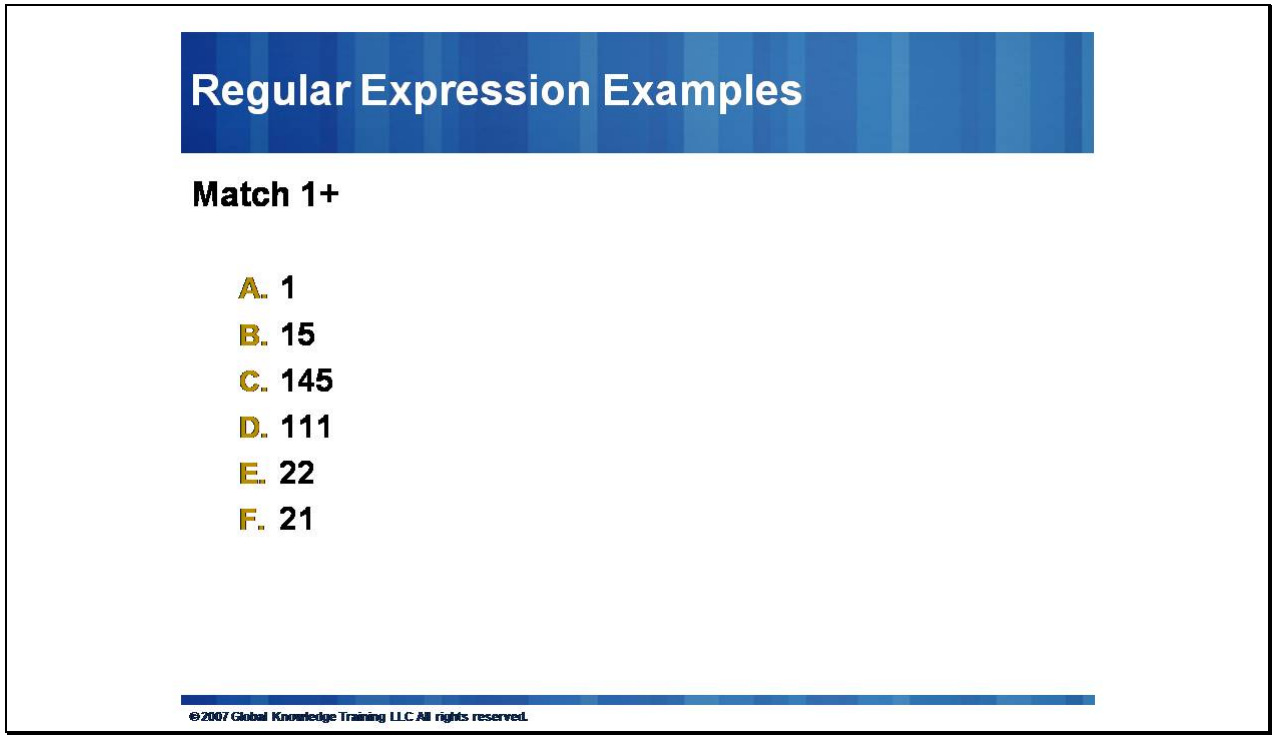
An atom is a single character or a grouping.

How do you match AS sequences “23 45” and “23 78 45” in a single regular expression?

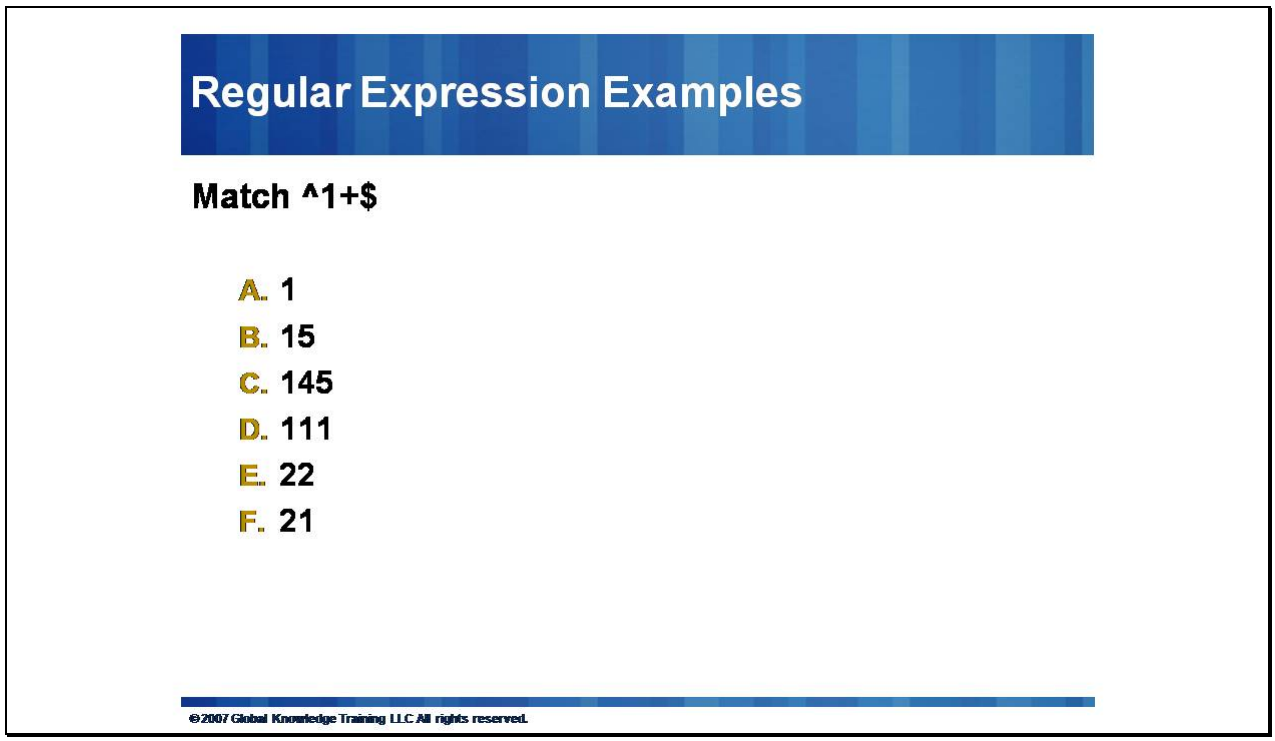
Answer: `_23(_78)?_45_`


Regular Expression Examples
Match ^1.$
A. 1
B. 15
C. 145
D. 111
E. 21
©2007 Global Knowledge Training LLC All rights reserved.

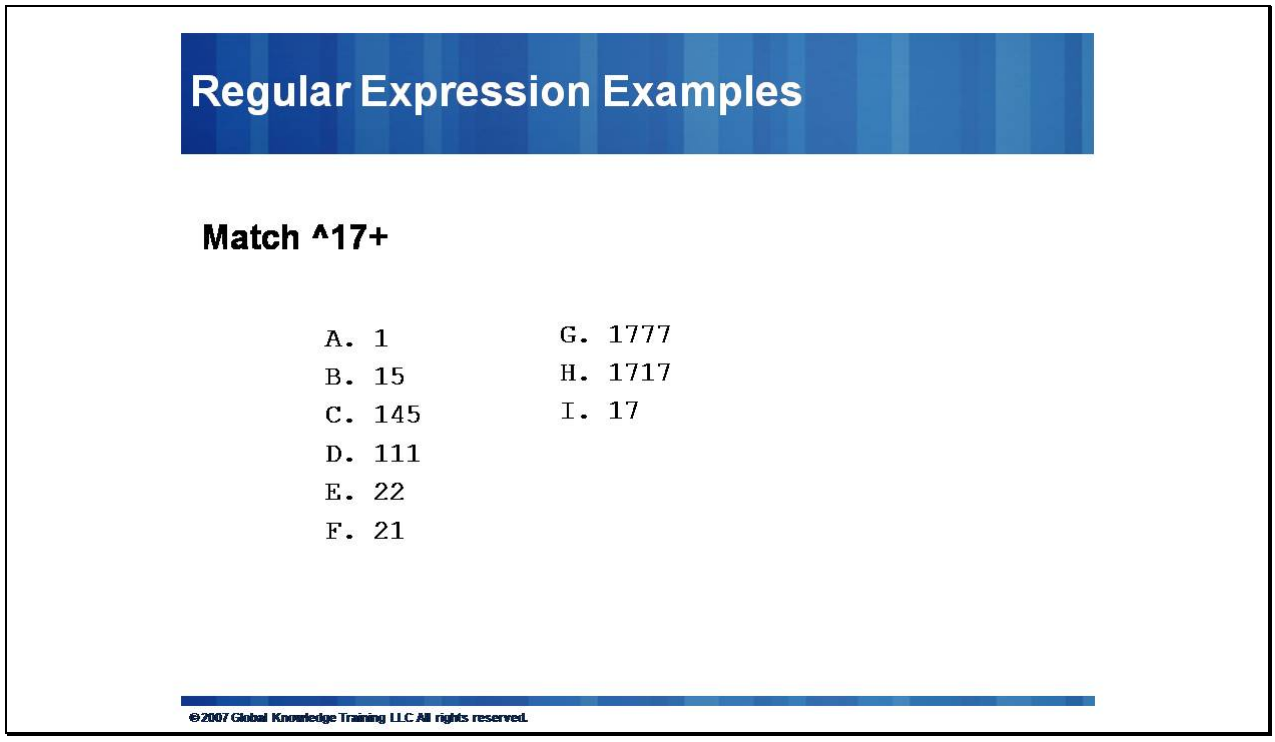
A, B


Regular Expression Examples
Match 1+
A. 1
B. 15
C. 145
D. 111
E. 22
F. 21
©2007 Global Knowledge Training LLC All rights reserved.

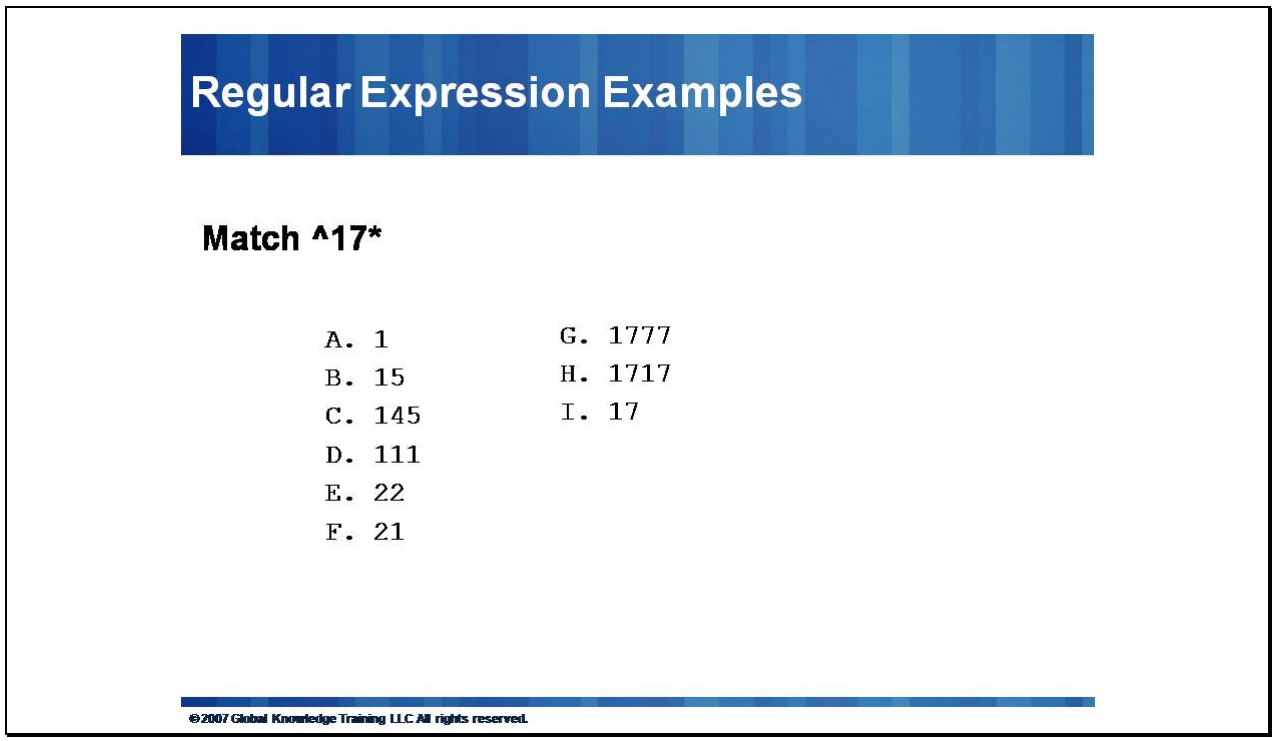
A, B, C, D, F


Regular Expression Examples
Match ^1+$
A. 1
B. 15
C. 145
D. 111
E. 22
F. 21
©2007 Global Knowledge Training LLC All rights reserved.

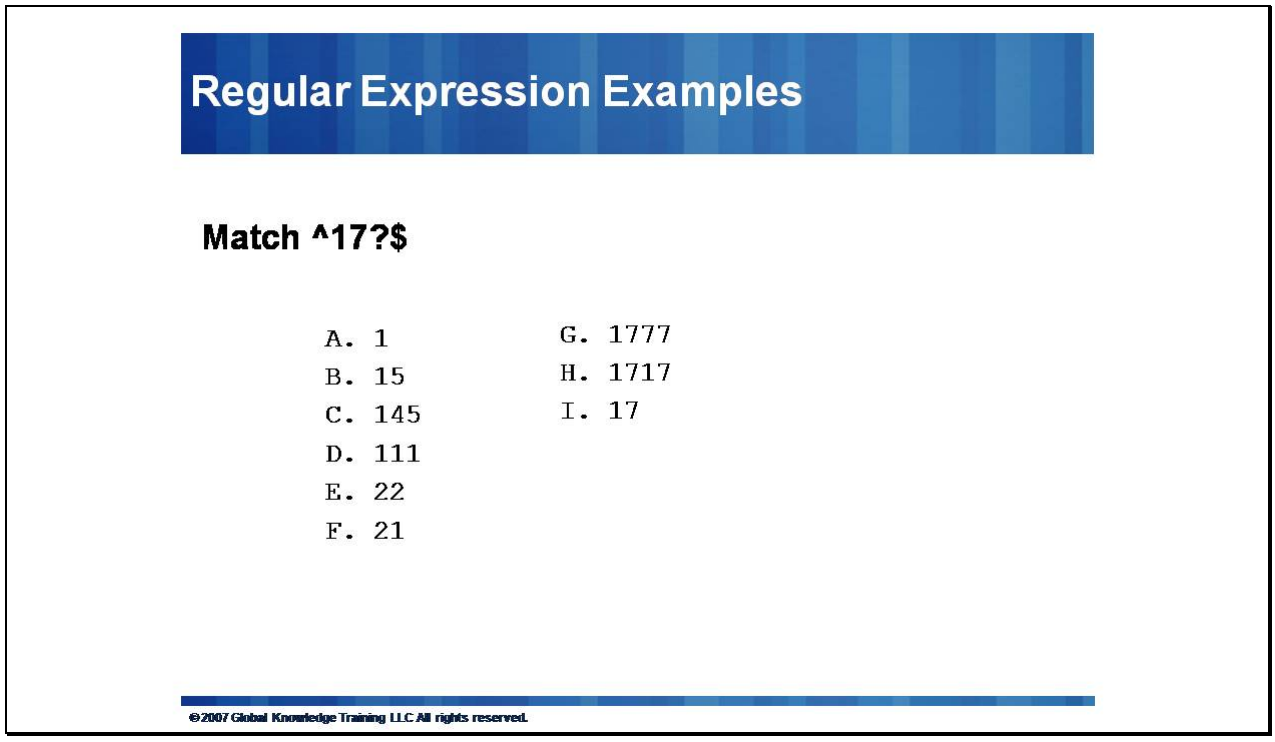
A, D


Regular Expression Examples
Match ^17+
A. 1
B. 15
C. 145
D. 111
E. 22
F. 21
G. 1777
H. 1717
I. 17
©2007 Global Knowledge Training LLC All rights reserved.

G, H, I


Regular Expression Examples
Match ^17*
A. 1
B. 15
C. 145
D. 111
E. 22
F. 21
G. 1777
H. 1717
I. 17
©2007 Global Knowledge Training LLC All rights reserved.

A, B, C, D, G, H, I


Regular Expression Examples
Match ^17?$
A. 1
B. 15
C. 145
D. 111
E. 22
F. 21
G. 1777
H. 1717
I. 17
©2007 Global Knowledge Training LLC All rights reserved.

A, I

String Matching

_100_	Going through AS 100
^100$	Directly connected to AS 100
_100$	Originated in AS 100
^100_.	Networks behind AS 100
^ [0-9]+$	AS paths one AS long
^([0-9]+)(_\1)*$	Prepending performed in neighboring originating AS
^$	Networks originated in local AS
.*	Matches everything

Configuring Prefix-Lists

```
router(config)#
ip prefix-list list-name [seq seq] {permit|deny}
network/len [ge value] [le value]
```

- **Prefix-lists have names and sequence numbers (like route-maps).**
- **An entry with no** le **or** ge **parameter matches exactly the specified prefix.**
- **An entry with an** le **or** ge **parameter matches any route within the address space of** *address/prefix* **with prefix longer or equal to** ge *value* **and shorter than or equal to** le *value*.

Configuring Prefix-Lists (Cont.)

- **Prefix-list matching rules**
  - **Prefix-list entries with no** ge **or** le **option match only the specified route.**
    - **Similar to IP access-lists with no wildcard bits**
    - **Matching also considers subnet mask**

Which of the following routes will be matched by:

```
ip prefix-list MyList permit 192.168.0.0/16?
```

✓ `192.168.0.0/16` X `192.168.0.0/20` X `192.168.2.0/24`

Configuring Prefix-Lists (Cont.)

- **A prefix-list entry with** ge **or** le **option matches any prefix within specified address space where the subnet mask falls within specified limits.**

Which of the following routes will be matched by:

```
ip prefix-list MyList permit 192.168.0.0/16 le 20?
```

✓ `192.168.0.0/16` ✓ `192.168.17.0/20` X `192.168.2.0/24`

```
ip prefix-list MyList permit 192.168.0.0/16 ge 18?
```

X `192.168.0.0/16` ✓ `192.168.17.0/20` ✓ `192.168.2.0/24`

Configuring Prefix-Lists (Cont.)

What will be matched by:

- **ip prefix-list A permit 0.0.0.0/0 ge 32**
- **ip prefix-list B permit 128.0.0.0/2 ge 17**
- **ip prefix-list C permit 0.0.0.0/0 le 32**
- **ip prefix-list D permit 0.0.0.0/0**
- **ip prefix-list E permit 0.0.0.0/1 le 24**

a) **All host routes**
b) **Any subnet in class B address space**
c) **All routes**
d) **Just the default route**
e) **Any prefix in class A address space covering at least 254 addresses**

Configuring Prefix-Lists (Cont.)

What will be matched by:

- **ipv6 prefix-list A permit ::/0 ge 128**
- **ipv6 prefix-list B permit FEC0::/10 ge 11**
- **ipv6 prefix-list C permit ::/0 le 128**
- **ipv6 prefix-list D permit ::/0**
- **ipv6 prefix-list E permit xxxx:xxxx:xxxx:xxxx::/64**

a) **All host routes**
b) **Any site local address space**
c) **All routes**
d) **Just the default route**
e) **A specific prefix with a length of 64 bits**

Configuring Outbound Route Filtering

```
router(config-router)#
neighbor ip-address capability orf prefix-list
[receive|send|both]
```

- This command enables negotiation of prefix-list ORF capability during session setup.
- The ORF-capable BGP speaker will install ORFs per neighbor.
- Option:
  - "Both" allows sending and receiving of prefix-lists.
  - "Send" allows only sending of prefix-lists.
  - "Receive" allows only receiving of prefix-lists.

Sample: Outbound Route Filtering

```
S5#sh run | b bgp
router bgp 200
 neighbor 208.28.1.129 remote-as 100
 neighbor 208.28.1.129 send-community both
 neighbor 208.28.1.129 capability orf prefix-list send
 neighbor 208.28.1.129 prefix-list ProbD in
!
ip prefix-list ProbD seq 5 deny 17.80.1.0/24
ip prefix-list ProbD seq 10 permit 0.0.0.0/0 le 32

R2#sh run | b bgp
router bgp 100
 bgp router-id 2.2.2.2
 neighbor 208.28.1.130 remote-as 200
 neighbor 208.28.1.130 capability orf prefix-list receive
!

R2#sh ip bgp neighbors 208.28.1.130 receive prefix-filter
Address family: IPv4 Unicast
ip prefix-list 208.28.1.130: 2 entries
   seq 5 deny 17.80.1.0/24
   seq 10 permit 0.0.0.0/0 le 32
```

BGP Link Bandwidth Feature

- **Used to enable multipath load balancing for external links with unequal bandwidth capacity**
- **Enabled under an IPv4 or VPNv4 address family sessions by entering the** bgp dmzlink-bw **command**
- **Routes learned from directly connected external neighbor propagated through the IBGP network with the bandwidth of the source external link**

Configuring BGP Link Bandwidth

- **bgp dmzlink-bw**

```
Router(config-router-af)# bgp dmzlink-bw
```

- **neighbor ip-address dmzlink-bw**

```
Router(config-router-af)# neighbor 172.16.1.1 dmzlink-bw
```

- **neighbor ip-address send-community [both | extended | standard]**

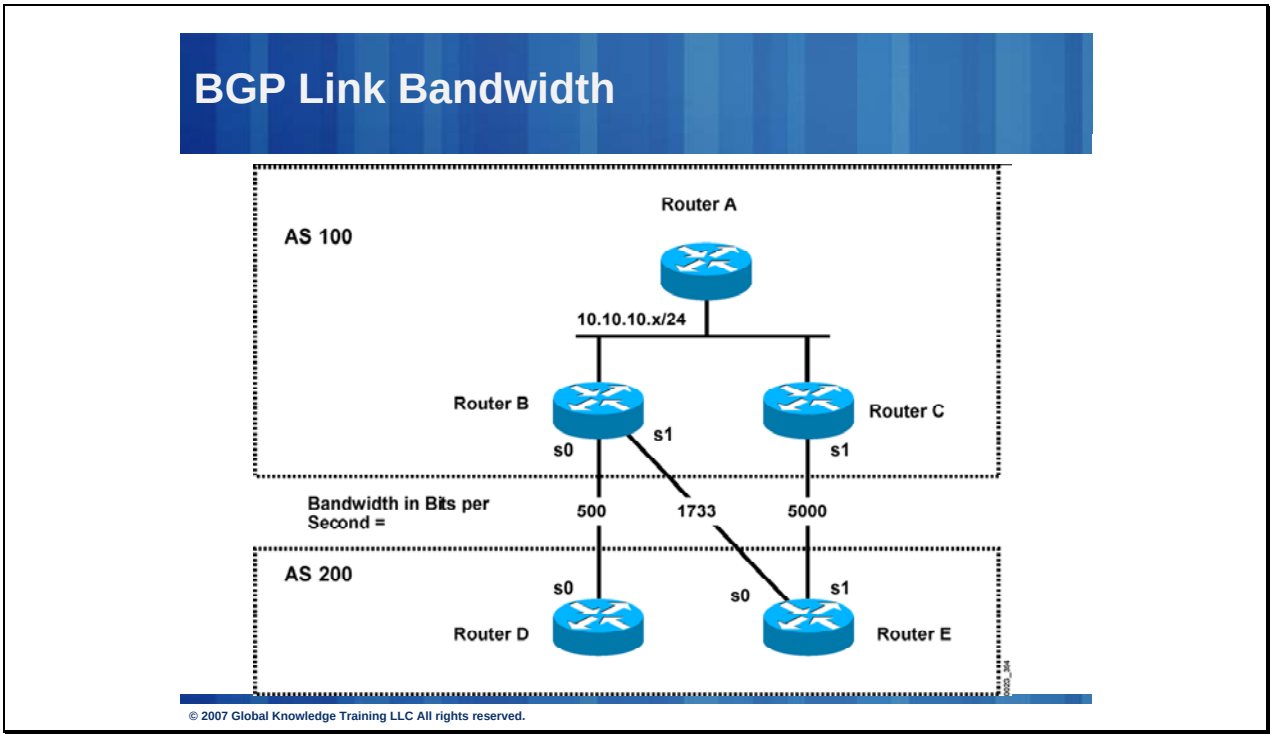
```
Router(config-router-af)# neighbor 10.10.10.1 send-community extended
```


The bgp dmzlink-bw configures BGP to distribute traffic proportionally to the bandwidth of the link. This command must be entered on each router that contains an external interface that is to be used for multipath load balancing.

The neighbor ip-address dmzlink-bw configures BGP to include the link bandwidth attribute for routes learned from the external interface specified IP address. This command must be configured for each eBGP link that is to be configured as a multipath.

Enabling this command allows the bandwidth of the external link to be propagated through the link bandwidth extended community.

The neighbor ip-address send-community [both | extended | standard] command must be configured for iBGP peers to which the link bandwidth extended community attribute is to be propagated.



BGP Link Bandwidth Configuration

Router A Configuration

In the following example, Router A is configured to support iBGP multipath load balancing and to exchange the BGP extended community attribute with iBGP neighbors:

Router A(config)# **router bgp 100**

Router A(config-router)# **neighbor 10.10.10.2 remote-as 100**

Router A(config-router)# **neighbor 10.10.10.2 update-source Loopback 0**

Router A(config-router)# **neighbor 10.10.10.3 remote-as 100**

Router A(config-router)# **neighbor 10.10.10.3 update-source Loopback 0**

Router A(config-router)# **address-family ipv4**

Router A(config-router)# **bgp dmzlink-bw**

Router A(config-router-af)# **neighbor 10.10.10.2 activate**

Router A(config-router-af)# **neighbor 10.10.10.2 send-community both**

Router A(config-router-af)# **neighbor 10.10.10.3 activate**

Router A(config-router-af)# **neighbor 10.10.10.3 send-community both**

Router A(config-router-af)# **maximum-paths ibgp 6**

Router B Configuration

In the following example, Router B is configured to support multipath load balancing, to distribute Router D and Router E link traffic proportionally to the bandwidth of each link, and to advertise the bandwidth of these links to iBGP neighbors as an extended community:

Router B(config)# **router bgp 100**

Router B(config-router)# **neighbor 10.10.10.1 remote-as 100**

Router B(config-router)# **neighbor 10.10.10.1 update-source Loopback 0**

Router B(config-router)# **neighbor 10.10.10.3 remote-as 100**

Router B(config-router)# **neighbor 10.10.10.3 update-source Loopback 0**

Router B(config-router)# **neighbor 172.16.1.1 remote-as 200**

Router B(config-router)# **neighbor 172.16.1.1 ebgp-multihop 1**

Router B(config-router)# **neighbor 172.16.2.2 remote-as 200**

Router B(config-router)# **neighbor 172.16.2.2 ebgp-multihop 1**

Router B(config-router)# **address-family ipv4**

Router B(config-router-af)# **bgp dmzlink-bw**

Router B(config-router-af)# **neighbor 10.10.10.1 activate**

Router B(config-router-af)# **neighbor 10.10.10.1 next-hop-self**

Router B(config-router-af)# **neighbor 10.10.10.1 send-community both**

Router B(config-router-af)# **neighbor 10.10.10.3 activate**

Router B(config-router-af)# **neighbor 10.10.10.3 next-hop-self**

Router B(config-router-af)# **neighbor 10.10.10.3 send-community both**

Router B(config-router-af)# **neighbor 172.16.1.1 activate**

Router B(config-router-af)# **neighbor 172.16.1.1 dmzlink-bw**

Router B(config-router-af)# **neighbor 172.16.2.2 activate**

Router B(config-router-af)# **neighbor 172.16.2.2 dmzlink-bw**

Router B(config-router-af)# **maximum-paths ibgp 6**

Router B(config-router-af)# **maximum-paths 6**

Router C Configuration

In the following example, Router C is configured to support multipath load balancing and to advertise the bandwidth of the link with Router E to iBGP neighbors as an extended community:

Router C(config)# **router bgp 100**

Router C(config-router)# **neighbor 10.10.10.1 remote-as 100**

Router C(config-router)# **neighbor 10.10.10.1 update-source Loopback 0**

Router C(config-router)# **neighbor 10.10.10.2 remote-as 100**

Router C(config-router)# **neighbor 10.10.10.2 update-source Loopback 0**

Router C(config-router)# **neighbor 172.16.3.30 remote-as 200**

Router C(config-router)# **neighbor 172.16.3.30 ebgp-multihop 1**

Router C(config-router)# **address-family ipv4**

Router C(config-router-af)# **bgp dmzlink-bw**

Router C(config-router-af)# **neighbor 10.10.10.1 activate**

Router C(config-router-af)# **neighbor 10.10.10.1 send-community both**

Router C(config-router-af)# **neighbor 10.10.10.1 next-hop-self**

Router C(config-router-af)# **neighbor 10.10.10.2 activate**

Router C(config-router-af)# **neighbor 10.10.10.2 send-community both**

Router C(config-router-af)# **neighbor 10.10.10.2 next-hop-self**

Router C(config-router-af)# **neighbor 172.16.3.3 activate**

Router C(config-router-af)# **neighbor 172.16.3.3 dmzlink-bw**

Router C(config-router-af)# **maximum-paths ibgp 6**

Router C(config-router-af)# **maximum-paths 6**

Verifying BGP Link Bandwidth

The examples in this section show the verification of this feature on Router A and Router B.

Router B

In the following example, the **show ip bgp** command is entered on Router B to verify that two unequal cost best paths have been installed into the BGP routing table. The bandwidth for each link is displayed with each route.

Router B# show ip bgp 192.168.1.0

BGP routing table entry for 192.168.1.0/24, version 48

Paths: (2 available, best #2)

Multipath: eBGP

Advertised to update-groups:

1 2

200

172.16.1.1 from 172.16.1.2 (192.168.1.1)

Origin incomplete, metric 0, localpref 100, valid, external, multipath, best

Extended Community: 0x0:0:0

DMZ-Link Bw 278 kbytes

200

172.16.2.2 from 172.16.2.2 (192.168.1.1)

Origin incomplete, metric 0, localpref 100, valid, external, multipath, best

Extended Community: 0x0:0:0

DMZ-Link Bw 625 kbytes

Router A

In the following example, the **show ip bgp** command is entered on Router A to verify that the link bandwidth extended community has been propagated through the iBGP network to Router A. The output shows that a route for each exit link (on Router B and Router C) to autonomous system 200 has been installed as a best path in the BGP routing table.

Router A# show ip bgp 192.168.1.0

BGP routing table entry for 192.168.1.0/24, version 48

Paths: (3 available, best #3)

Multipath: eBGP

Advertised to update-groups:

1 2

200

172.16.1.1 from 172.16.1.2 (192.168.1.1)

Origin incomplete, metric 0, localpref 100, valid, external, multipath

Extended Community: 0x0:0:0

DMZ-Link Bw 278 kbytes

200

172.16.2.2 from 172.16.2.2 (192.168.1.1)

Origin incomplete, metric 0, localpref 100, valid, external, multipath, best

Extended Community: 0x0:0:0

DMZ-Link Bw 625 kbytes

200

172.16.3.3 from 172.16.3.3 (192.168.1.1)

Origin incomplete, metric 0, localpref 100, valid, external, multipath, best

Extended Community: 0x0:0:0

Removing Private AS Numbers

```
router(config-router)#
neighbor ip-address remove-private-as
```

- The command modifies AS-path processing on outgoing updates sent to specified neighbor.
- Private AS numbers are removed from the tail of the AS path before the update is sent.
- Private AS numbers followed by a public AS number are not removed.
- The AS number of the sender is prepended to the AS path after this operation.

AS Number Translation

```
router(config-router)#
neighbor ip-address local-as local-as no-prepend replace-as
```

- Optionally, the customer can get two different private AS numbers assigned by the service providers.
- Internally, the customer can use an ISP-assigned AS number or even any other private AS number.
- Externally, the customer is seen as one private AS number to ISP 1 and as a different AS to ISP 2.
- Note: When you are using this option, the AS path of the customer network contains two AS numbers. The ISP has to adapt the incoming AS-path filters.

Network Design with Route Reflectors

- **Route reflector rules**
  - Route reflector rules divide a transit AS into smaller areas (called clusters).
  - Each cluster contains route reflectors and route reflector clients.
  - Routers that do not support route reflector functionality act as a one-router cluster or as a route reflector client.
- **IBGP session rules**
  - All clients in a cluster must establish IBGP sessions with and only with all route reflectors in the cluster.
  - An IBGP full mesh between all route reflectors within the AS is required.
  - Routers that are not route reflectors can participate in the IBGP full mesh or be route reflector clients.
- **Route reflector reflects**
  - A route reflector will reflect from:
  - client to client
  - non-client to client
  - client to non-client

Configuring Route Reflectors

```
router(config-router)#
bgp cluster-id cluster-id
```

- Optionally assigns a cluster-ID to the route reflector (default value is router-ID)
- Required only for clusters with redundant reflectors
- Cluster-ID cannot be changed after the first client is configured

```
router(config-router)#
neighbor ip-address route-reflector-client
```

- Configures an IBGP neighbor to be a client of this reflector

Configuring Route Reflectors

router(config-router)#
bgp cluster-id cluster-id

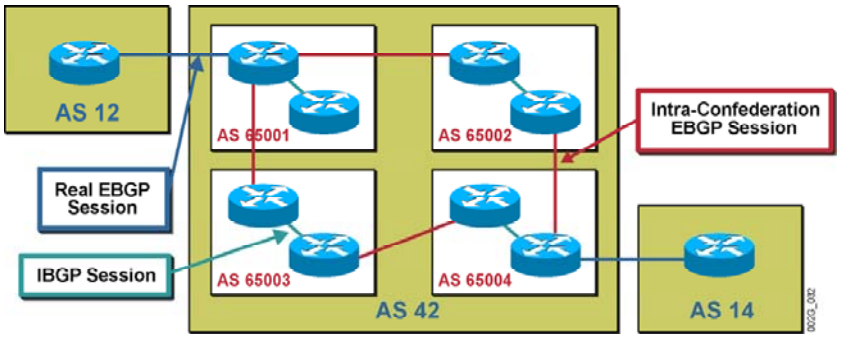
- Optionally assigns a cluster-ID to the route reflector (default value is router-ID)
- Required only for clusters with redundant reflectors

router(config-router)#
neighbor ip-address route-reflector-client

- Configures an IBGP neighbor to be a client of this reflector

©2007 Global Knowledge Training LLC All rights reserved.

BGP Confederations



- Splitting the AS into smaller autonomous systems would reduce the number of BGP sessions, but extra AS numbers are not available.
- Confederations enable internal AS numbers to be hidden and announce only one (external) AS number to EBGP neighbors.

Intra-Confederation

- Behaves like EBGP session during session establishment
  - The EBGP neighbor has to be directly connected, or you have to configure ebgp-multihop on the neighbor.
- Behaves like IBGP session when propagating routing updates
  - The local preference, MED, and next-hop attributes are retained.
  - The whole confederation can run one IGP, providing optimal routing based on the next-hop attribute in the BGP routing table.

BGP Confederation Design Rules

- IBGP full mesh within each member-AS is required.
  - Route reflectors might be used within each AS to relax the IBGP full-mesh requirements.
- There is no topology limitation on EBGP sessions between autonomous systems within a confederation.
  - The intra-confederation EBGP sessions will follow the physical topology of the network.

Configuring BGP Confederations

```
router(config)#
router bgp member-as-number
```

- Remove the old BGP process and configure BGP process with member-AS number

```
router(config-router)#
bgp confederation identifier external-as-number
```

- Configures external confederation-wide AS number

```
router(config-router)#
bgp confederation peers list-of-intra-confederation-as
```

- Defines all the other autonomous systems in the confederation

BGP Scan Time

```
router(config-router)#
bgp scan-time scanner-interval
```

- This command changes the default value of BGP scanner process runs (default = 60 seconds).
- The BGP scanner walks the BGP table and confirms the reachability of next hops.
- The BGP scanner process is also responsible for advanced features such as conditional advertisement check and performing route dampening.

BGP Advertisement Interval

```
router(config-router)#
neighbor {ip-address | peer-group-name} advertisement-
interval seconds
```

- This command changes the default time interval in the sending of BGP routing updates for a specific neighbor:
  - If lowered, can improve convergence
  - Can consume considerable resources in a jittery network if the value is set too low
- Default values:
  - 30 seconds for EBGP neighbors
  - 5 seconds for IBGP neighbors

BGP Peer Group Limitations

- Peer groups have a number of limitations because of the way that they are used to build BGP updates.
  - Per-neighbor BGP parameters that affect outbound updates cannot be changed for peer group members.
  - IBGP and EBGP neighbors cannot be mixed in a peer group.

Configuring BGP Route Dampening

```
router(config-router)#
bgp dampening [half-life reuse suppress max-suppress-time] [route-map map-name]
```

- Configures BGP route dampening
- BGP dampening parameters:
  - *half-life* — Decay time in which the penalty is halved
  - *suppress* — Value when the route starts dampening
  - *reuse* — Value when the dampened route is reused
  - *max-suppress-time* — Maximum time to suppress the route
  - route-map — Name of route-map controlling dampening

Configuring BGP Route Dampening

- Most Internet service providers use default values:
  - A flapping route is dampened after three successive flaps.
  - A route stays suppressed for approximately 30 minutes.
  - Net result: The route is lost for 30 minutes if a BGP session with a neighbor is cleared three times in succession.
  - Default dampening parameter values are:
    - half-life 15 minutes
    - suppress 2000
    - reuse 750
    - max-suppress-time 60 minutes (4x half-life)
    - per-flap penalty 1000 (nonconfigurable)

Configuring BGP Route Dampening

```
router(config-route-map)#
set dampening half-life reuse suppress max-suppress-time
```

- This command sets the BGP dampening parameters for individual routes matched by a route-map entry.
- Apply this route-map to the bgp dampening command instead of specifying individual parameters.
- Applications:
  - Less aggressive dampening of routes toward root DNS servers (or other servers)
  - Dampening of smaller prefixes more aggressively
  - Selective dampening based on BGP neighbor and route-map match criteria

Releasing Dampened Routes

```
router#
clear ip bgp ip-address flap-statistics [{regexp regexp} |
{filter-list list-name} | {ip-address network-mask}]
```

- Clears the flap statistics but does not release dampened routes

```
router#
clear ip bgp dampening [ip-address network-mask]
```

- Releases all the dampened routes or just the specified network
- Flap statistics or dampened routes also cleared when the BGP session with the neighbor is lost