



Università degli Studi Roma Tre  
Dipartimento di Informatica e Automazione  
Computer Networks Research Group

# netkit lab

## bgp: simple-peering

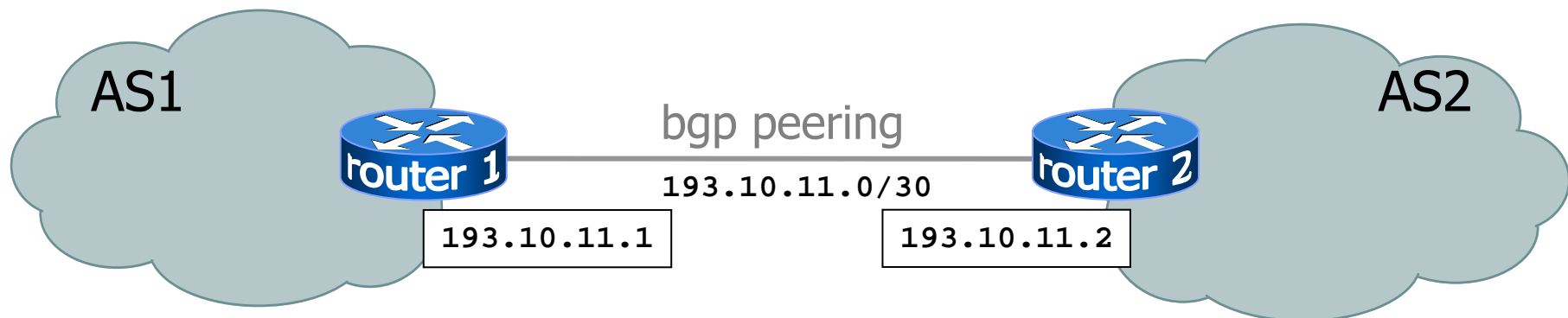
<b>Version</b>	2.0
<b>Author(s)</b>	G. Di Battista, M. Patrignani, M. Pizzonia, F. Ricci, M. Rimondini
<b>E-mail</b>	contact@netkit.org
<b>Web</b>	<a href="http://www.netkit.org/">http://www.netkit.org/</a>
<b>Description</b>	setting up a bgp peering between two autonomous systems

# copyright notice

- All the pages/slides in this presentation, including but not limited to, images, photos, animations, videos, sounds, music, and text (hereby referred to as "material") are protected by copyright.
- This material, with the exception of some multimedia elements licensed by other organizations, is property of the authors and/or organizations appearing in the first slide.
- This material, or its parts, can be reproduced and used for didactical purposes within universities and schools, provided that this happens for non-profit purposes.
- Information contained in this material cannot be used within network design projects or other products of any kind.
- Any other use is prohibited, unless explicitly authorized by the authors on the basis of an explicit agreement.
- The authors assume no responsibility about this material and provide this material "as is", with no implicit or explicit warranty about the correctness and completeness of its contents, which may be subject to changes.
- This copyright notice must always be redistributed together with the material, or its portions.

# a bgp peering between two ases

- bgp allows routers to exchange information only if a peering session is up
- a bgp peering is the tcp connection over which routing information will be exchanged



# peering configuration commands

command syntax

```
! <a-comment-on-a-single-line>
```

command syntax

```
router bgp <my-as-number>
```

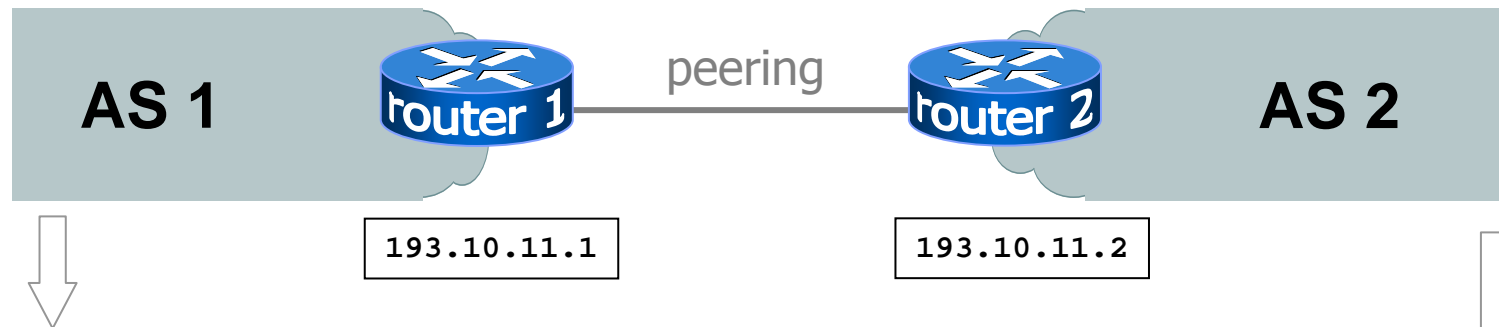
command syntax

```
neighbor <neighbor-ip> remote-as <neighbor-as-num>
```

command syntax

```
neighbor <neighbor-ip> description <text>
```

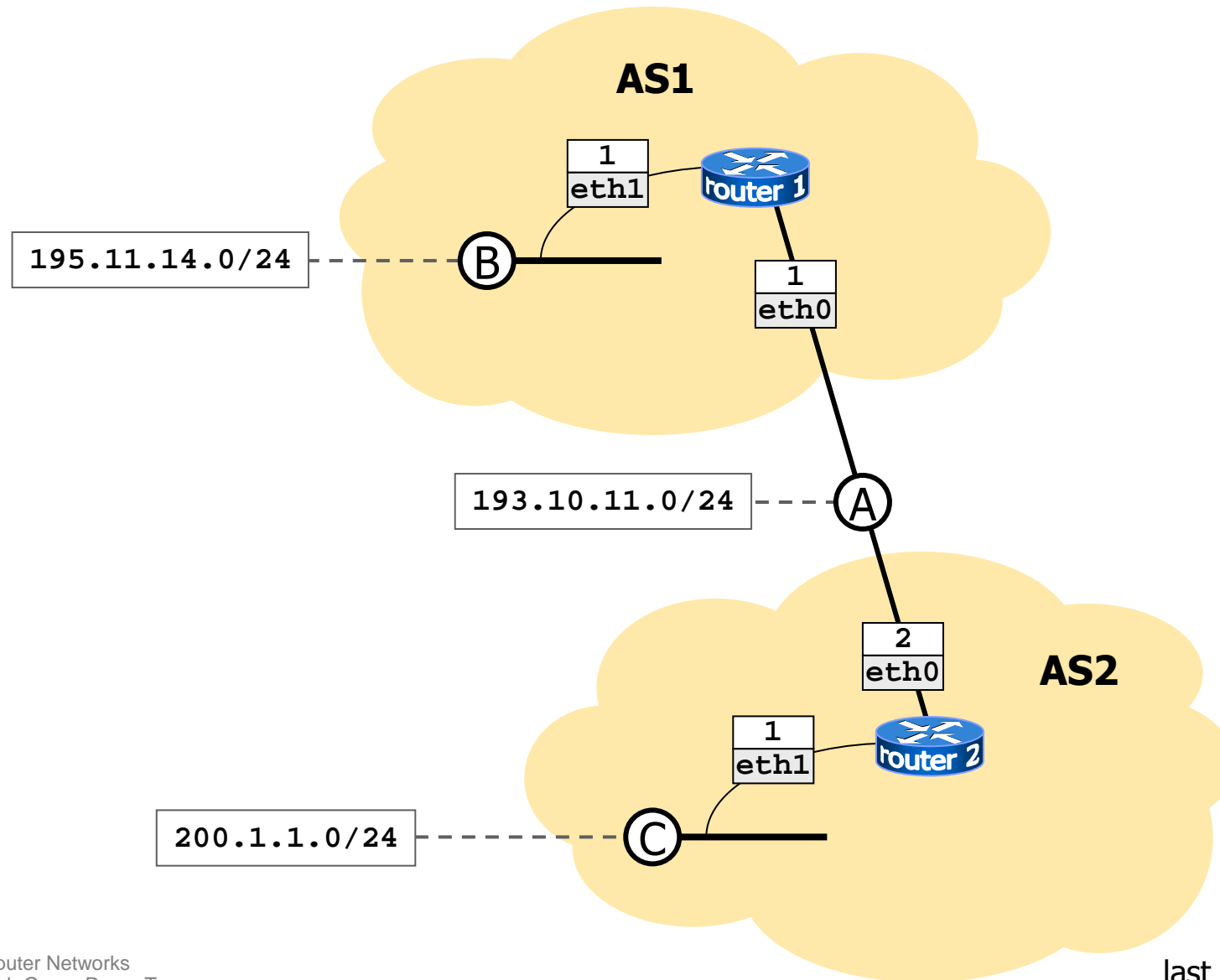
# peering configuration example



```
! router 1 configuration file
router bgp 1
neighbor 193.10.11.2 remote-as 2
neighbor 193.10.11.2 description Router 2
```

```
! router 2 configuration file
router bgp 2
neighbor 193.10.11.1 remote-as 1
neighbor 193.10.11.1 description Router 1
```

# a simple peering



# a simple peering

- launch the script

```
host machine
user@localhost:~$ cd netkit-lab_bgp-simple-peering
user@localhost:~/netkit-lab_bgp-simple-peering$ ./start
```

- check the bgpd configuration file

```
router1
router1:~# less /etc/zebra/bgpd.conf
!
hostname bgpd
password zebra
enable password zebra
!
router bgp 1
/etc/zebra/bgpd.conf
```

# a simple peering

- check the routing tables

```
router1:~# route
Kernel IP routing table
Destination      Gateway          Genmask          Flags Metric Ref    Use Iface
193.10.11.0      *               255.255.255.0   U        0      0      0 eth0
195.11.14.0      *               255.255.255.0   U        0      0      0 eth1
router1:~# █
```

- as no routing protocol (not even bgp!) is propagating routing information, only local routes are known

# a simple peering

- check the log file of the bgp daemon

router1

```
router1:~# less /var/log/zebra/bgpd.log
2007/05/22 11:01:06 BGP: BGPd 0.94 starting: vty@2605, bgp@179
2007/05/22 11:01:14 BGP: 193.10.11.2 [FSM] Timer (start timer expire).
2007/05/22 11:01:14 BGP: 193.10.11.2 [FSM] BGP_Start (Idle->Connect)
2007/05/22 11:01:14 BGP: 193.10.11.2 went from Idle to Connect
2007/05/22 11:01:14 BGP: 193.10.11.2 [Event] Connect start to 193.10.11.2 fd 9
2007/05/22 11:01:14 BGP: 193.10.11.2 [FSM] Non blocking connect waiting result
2007/05/22 11:01:17 BGP: 193.10.11.2 [Event] Connect failed (Operation now in
progress)
2007/05/22 11:01:17 BGP: 193.10.11.2 [FSM] TCP_connection_open_failed
(Connect->Active)
2007/05/22 11:01:17 BGP: 193.10.11.2 went from Connect to Active
2007/05/22 11:01:31 BGP: [Event] BGP connection from host 193.10.11.2
2007/05/22 11:01:31 BGP: [Event] Make dummy peer structure until read Open
packet
/var/log/zebra/bgpd.log
```

# a simple peering

- check the command line interface of bgpd

```
router1
router1:~# telnet localhost bgpd
Trying 127.0.0.1...
Connected to router1.
Escape character is '^]'.

Hello, this is zebra (version 0.94).
Copyright 1996-2002 Kunihiro Ishiguro.

User Access Verification

Password: zebra
bgpd> show ip bgp summary
BGP router identifier 195.11.14.1, local AS number 1
0 BGP AS-PATH entries
0 BGP community entries

Neighbor      V    AS MsgRcvd MsgSent  TblVer  InQ  OutQ Up/Down  State/PfxRcd
193.10.11.2   4    2     53     55      0    0    0 00:53:00      0

Total number of neighbors 1
bgpd> █
```

# a simple peering

- check the peering status

```
router1
bgpd> show ip bgp neighbors
BGP neighbor is 193.10.11.2, remote AS 2, local AS 1, external link
  Description: Router 2 of AS2
  BGP version 4, remote router ID 200.1.1.1
  BGP state = Established, up for 00:57:51
  Last read 00:00:49, hold time is 180, keepalive interval is 60 seconds
  Neighbor capabilities:
    Route refresh: advertised and received (old and new)
    Address family IPv4 Unicast: advertised and received
  Received 58 messages, 0 notifications, 0 in queue
  Sent 60 messages, 0 notifications, 0 in queue
  Route refresh request: received 0, sent 0
  Minimum time between advertisement runs is 30 seconds

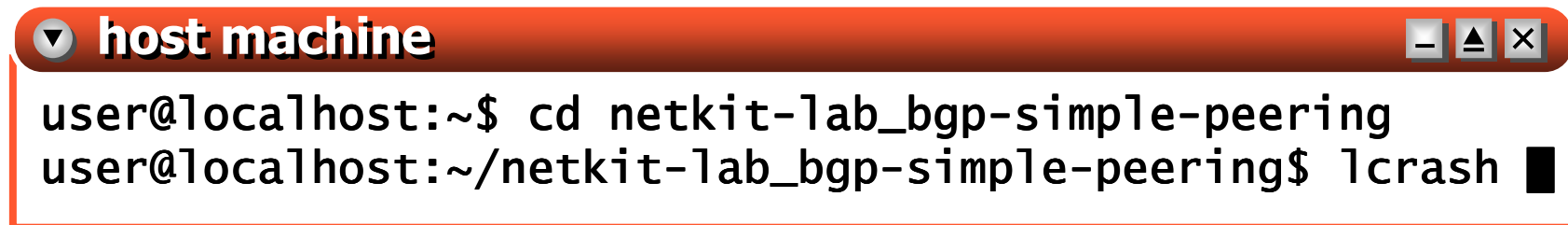
For address family: IPv4 Unicast
  Community attribute sent to this neighbor (both)
  0 accepted prefixes

  Connections established 1; dropped 0
  Local host: 193.10.11.1, Local port: 179
  Foreign host: 193.10.11.2, Foreign port: 3452
  Nexthop: 193.10.11.1
  Nexthop global: fe80::fcfd:c1ff:fe0a:b01
  Nexthop local: ::
  BGP connection: non shared network
  Read thread: on Write thread: off

bgpd> █
```

# a simple peering

- stop the lab



```
host machine
user@localhost:~$ cd netkit-lab_bgp-simple-peering
user@localhost:~/netkit-lab_bgp-simple-peering$ 1crash
```