BGP Filtering—Myths Legends and Reality:
Peer Filtering in the Modern Backbone

NANOG 35
October 24, 2005

Jim Deleskie, Teleglobe
Alin Popescu, Renesys
Tom Scholl, SBC Internet
Todd Underwood, Renesys

Overview

- Why filter?
- Why not just IRR?
- Novel technique described and evaluated
- Router performance under large prefix lists evaluated
- Conclusions

Why Filter All Peers?

- AS7007
- AS9121 <http://nanog.org/mtg-0505/underwood.html>
- 12/8 in Bolivia?
- Dozens of new /8s in China/Latvia?
- Route Hijackings/Black Holings
- Long-term untenability of transitively trusting all peers' customers.

Why Not filter All Peers?

Most (All) Large Providers Do Not Filter Peers.

Common Reasons:

- No accurate source of filter lists (IRRs not up to date)
- Configurations too large for routers
  - Too hard to manage/load/replace
  - CPU/Memory resources to run (especially during convergence)
- Cost/Benefit trade-off not worth it

Accurate Lists: Just Use IRRs

- Internet Routing Registries were designed specifically for this purpose
- Every network maintains (out of band) a list of every prefix they announce, and every network adjacency they maintain
- Prefix list filters are simply the union of the prefixes of every transitive downstream of the peer in the IRR

IRR as Elusive Holy Grail

- Complete, accurate IRR data does not exist
  - Some registries better than others, but nowhere near operationally useful anywhere
- Until people prefix-filter based on IRR data, it will not be maintained. Until accurate data exists, it will not be used for prefix filters. Chicken; Egg.
- A practical technique to supplement IRR data is necessary
- Will drive more use of the IRRs.

•Inaccurate / Incomplete IRR Data

One network (prospective customer):

- Partial Routes registered in 3 Registries
- AS-XXXX : APNIC (2000-10-23)
  - 274 advertised routes are registered (51%)
- AS-XXXXX : SAAVIS (2003-05-27)
  - Does not include own AS in route set !!!
  - 198 advertised routes are registered (37%)
- AS-XXXXX : VERIO (2000-08-28)
  - 278 advertised routes are registered (52%)
- 538 routes observed behind 6 current peers
  - Must filter one on something other then IRR or not filter at all

(Names withheld to protect the 'innocent'.)

Approach

- Attempt to debunk the two major objections to filtering all peers:
  - No possible source of filter lists
  - Routers can't handle the load
- Identify a set of novel techniques for validating filter lists without up-to-date IRRs; evaluate
- Test configs in relatively real-world test-beds
- Identify hardware/platform weaknesses

Overview: Peer Routes Validated

- Start with three things:
  - One or more somewhat-trusted IRRs
    - Assume incomplete
    - Assume some stale data
  - Peer routes from the target peer over a period of time
    - Trust some consensus over some period of time
    - Filter only deltas from that set
  - A Routeviews-style peer set with diverse, global full tables

Validate New Advertisement

- General Principle: routes seen stably by lots of peers over long time are assumed valid.
- Additional Observation: Delaying acceptance of new originations may be better than accepting illicit paths
- Four non-orthogonal dimensions:
  - Registry information
  - Origination
  - AS path
  - Peer confirmation (widely believed/routed)

Validate New Advertisement (cont.)

- **Bad Origin:** if origination is new for this peer and cannot be confirmed in an IRR or by many other peers, REJECT.
- **Invalid Path:** if the AS path contains invalid edges (unconfirmed adjacencies or directionality from global or peer data), REJECT.
- **Poor Peer Confirmation:** If only few peers validate origination or path, SUSPICIOUS.
- **Default:** ACCEPT.
  *(More important to implement **some** filtering now without discarding any legitimate traffic)*

Results

Took peer routes sent to a large European-based network from the following ASes over two weeks:

```
12956 1668 2914 3257 3356 3561 6453 6762 702
```

- Trusted first table
- Daily rate novel route median <0.5% (max ~3%)
- <12% novel routes are SUSPICIOUS (hard to categorize automatically) using naïve techniques
- Techniques work well for normal peering sessions
- Needs further testing against anomalous sessions (big leaks, hijackings)

Peer Route Validation Examples

1. (Easy) 222.124.42.0/24 3561 7473 7713 17974
   The path (for a different prefix) seen the previous day by this peer and the origination by AS17974 confirmed over time and across peers
2. (Harder) MULTIPLE ORIGINS:
   `149.43.0.0/16 1668 11351`
   Stable origination by 11351
   Previously also originated by 701
   Colgate multi-homing a prefix without an ASN.
   Stable, multiple originations can be allowed.

•Peer Route Validation Examples (cont)

4. (Hard) NOT SEEN:
   `194.187.56.0/22  702 12883 13249 15497 35409 35362`
   Routes from the previous day (assumed valid):
   `193.41.160.0/22 702 12883 13249 15497 35409 29327`
   `193.223.98.0/24 702 12883 13249 15497 35409`
   Edge 35409 35362 not seen by this peer, but confirmed by 80% of other peers
5. (Really Hard) MULTIPLE ORIGINS:
   `83.210.95.0/24 3356 4716 23918`
   Connexion by Boeing. Also originated by 23918, 33697, 30533, 31050, 29257
   Stretches the flexibility of multiple, stable originations

Prevent Recent Bad Events?

- AS9121 leak?
  Rejected for Origination
- AS26210 and AS12676 originating 12/8?
  Rejected for Origination
- Most Hijackings?
  Rejected for Origination
- Hijackings with sophisticated forged AS paths?
  Some rejected for path validation
  Some may be accepted

Proof of Concept Stage

- Plausible, but difficult and requires further work
- Acceptance of novel originations will be delayed unless registered in trusted IRR
  - Drives acceptance of IRRs
  - Modest delay (tunable) of routing of new originations
- Easier/cheaper than sBGP/soBGP?
  - Almost certainly
  - Does not accomplish all same goals, though
- 20-40% of transit ASes do not currently filter all customer sessions. Do that first.

Router Configuration Approach

- Filter prefixes per-peer
- No AS-path filtering yet (regexps believed to be too inefficient in current implementations)
- Considered a “one true list” prefix+AS-path approach, but discarded

Testing Overview

- 9 eBGP multihop sessions (No MD5)
- Three types of tests ran for both baseline and with prefix-filters enabled.
  - BGP Establishing – Sessions up 5 minutes after start of test.
  - BGP Operating – Sessions already up and idle for 20 minutes.
  - BGP Teardown – Sessions torn down 5 minutes after start of test.
- Each test 20 minutes (1200 seconds)

Testing Overview (cont)

- BGP Neighbor #1 – 2,324 prefixes
- BGP Neighbor #2 – 1,687 prefixes
- BGP Neighbor #3 – 20,075 prefixes
- BGP Neighbor #4 – 2,951 prefixes
- BGP Neighbor #5 – 32,087 prefixes
- BGP Neighbor #6 – 32,100 prefixes
- BGP Neighbor #7 – 9,381 prefixes
- BGP Neighbor #8 – 4,619 prefixes
- BGP Neighbor #9 – 8,627 prefixes
- Total: 113,851 prefixes

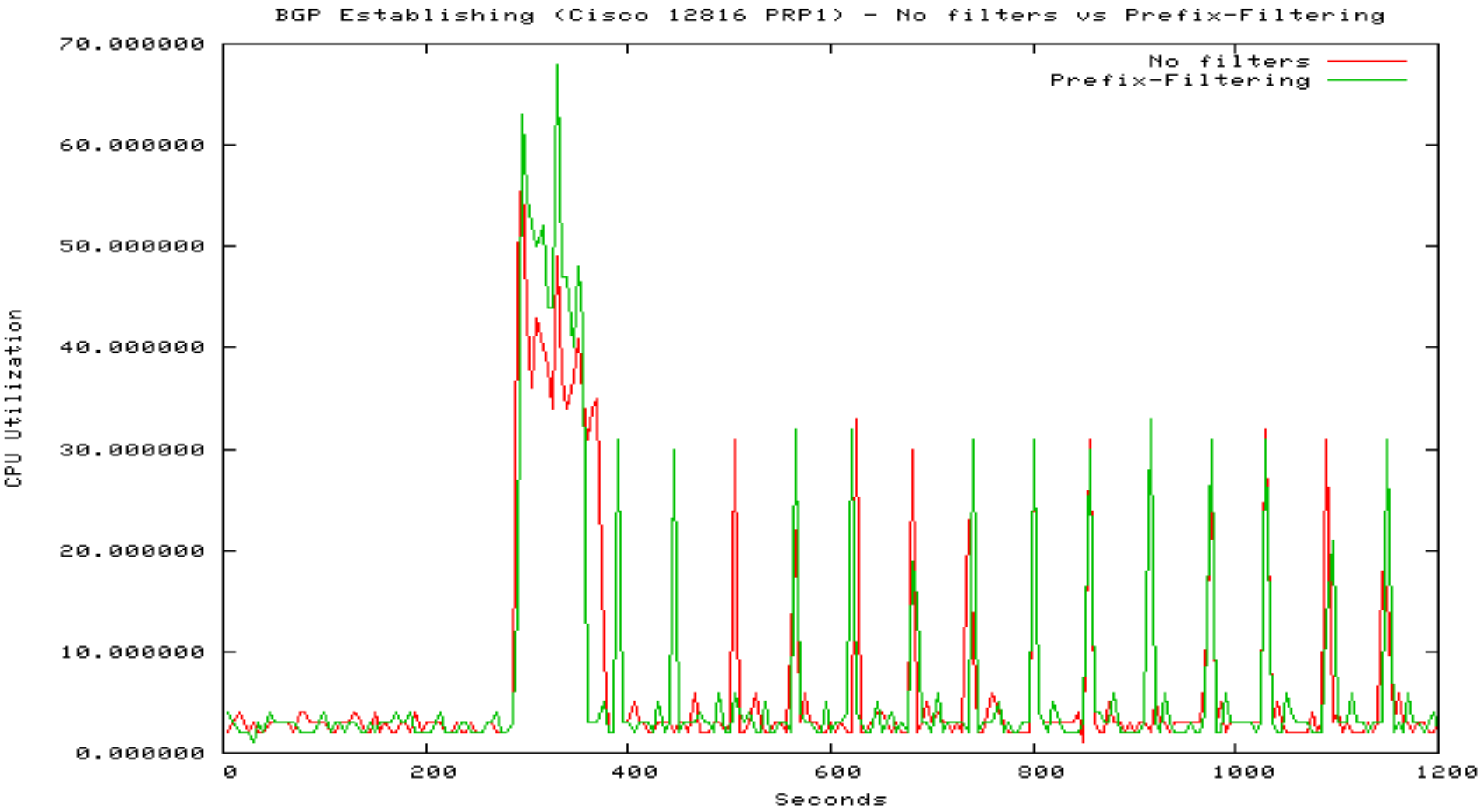
Testing Overview (cont)

- Three vendors tested
- Cisco GSR:
  - Cisco GSR 12410 – GRP
  - Cisco GSR 12816 – PRP1
  - Cisco GSR 12816 – PRP2
  - Cisco Catalyst 6500 Sup720
- Alcatel 7750 SR
- Juniper (RE-4.0)

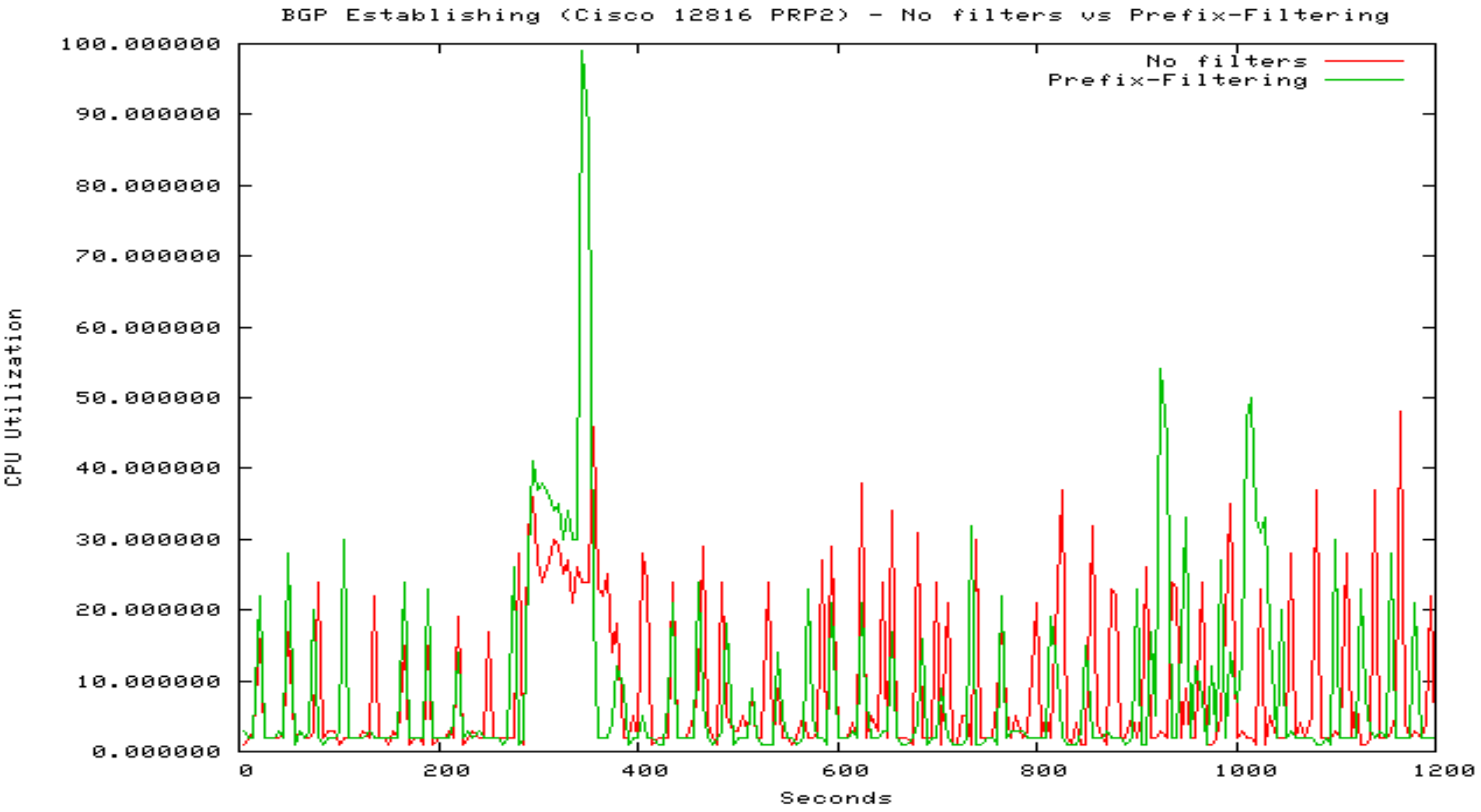
Testing Results - Cisco

- Configuration size:
  - Before prefix-lists:
    - 71,891 bytes uncompressed, 24,230 compressed
  - After prefix-lists:
    - 6,298,764 bytes uncompressed, 1,036,796 compressed
  - Typical configuration upload time: ~45-50 seconds
  - Prefix-List configuration was ~5.1MB
  - **GRP-B NVRAM size -> 507k – will NOT work**
  - PRP-1 NVRAM size -> 2043k – OK
  - PRP-2 NVRAM size -> 2043k – OK
  - Catalyst 6500
    - Sup720 NVRAM size -> 1900k – OK
    - **Sup2/MSFC2 NVRAM size -> 128K – will NOT work**

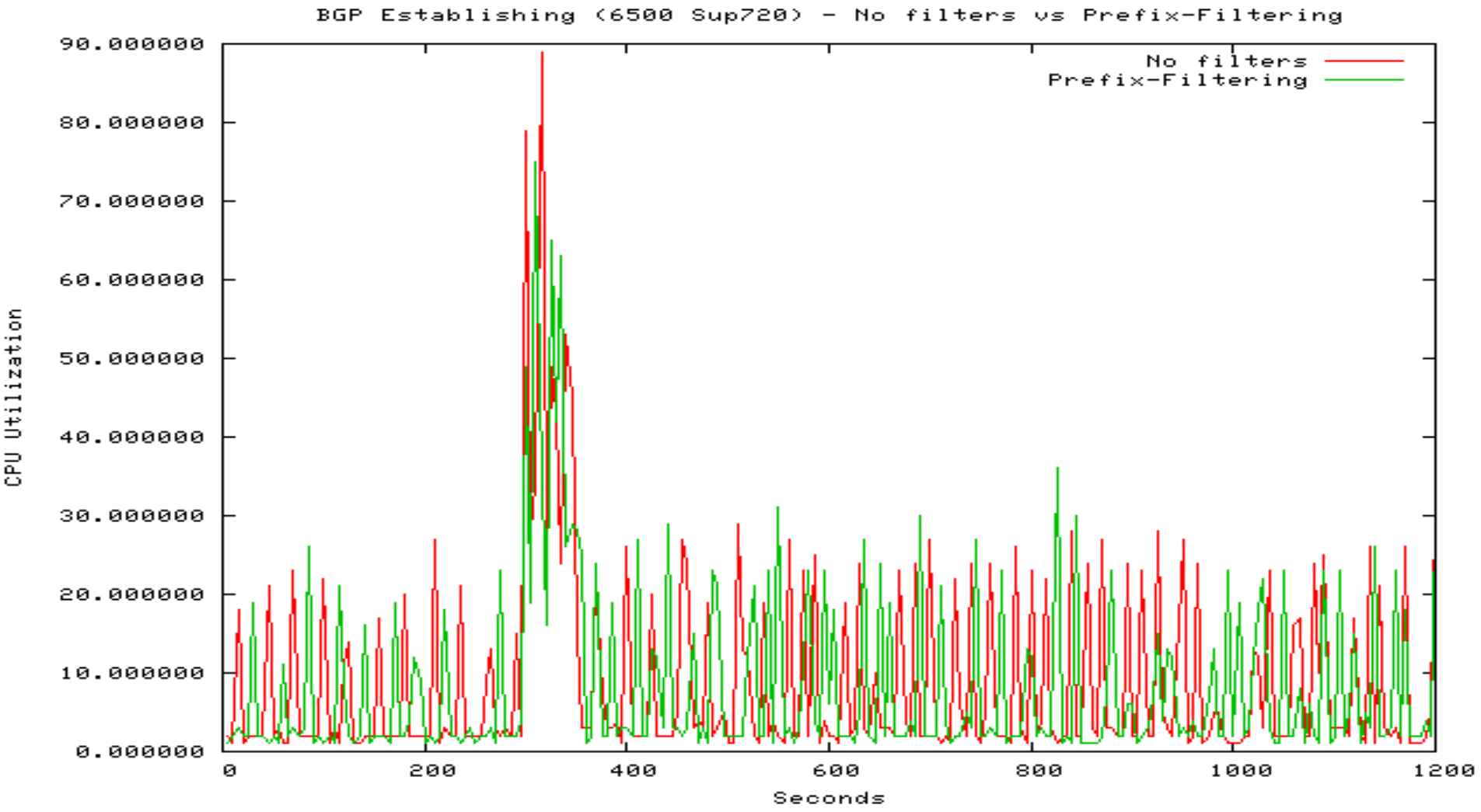
Testing Results – Cisco - PRP1



Testing Results – Cisco - PRP2



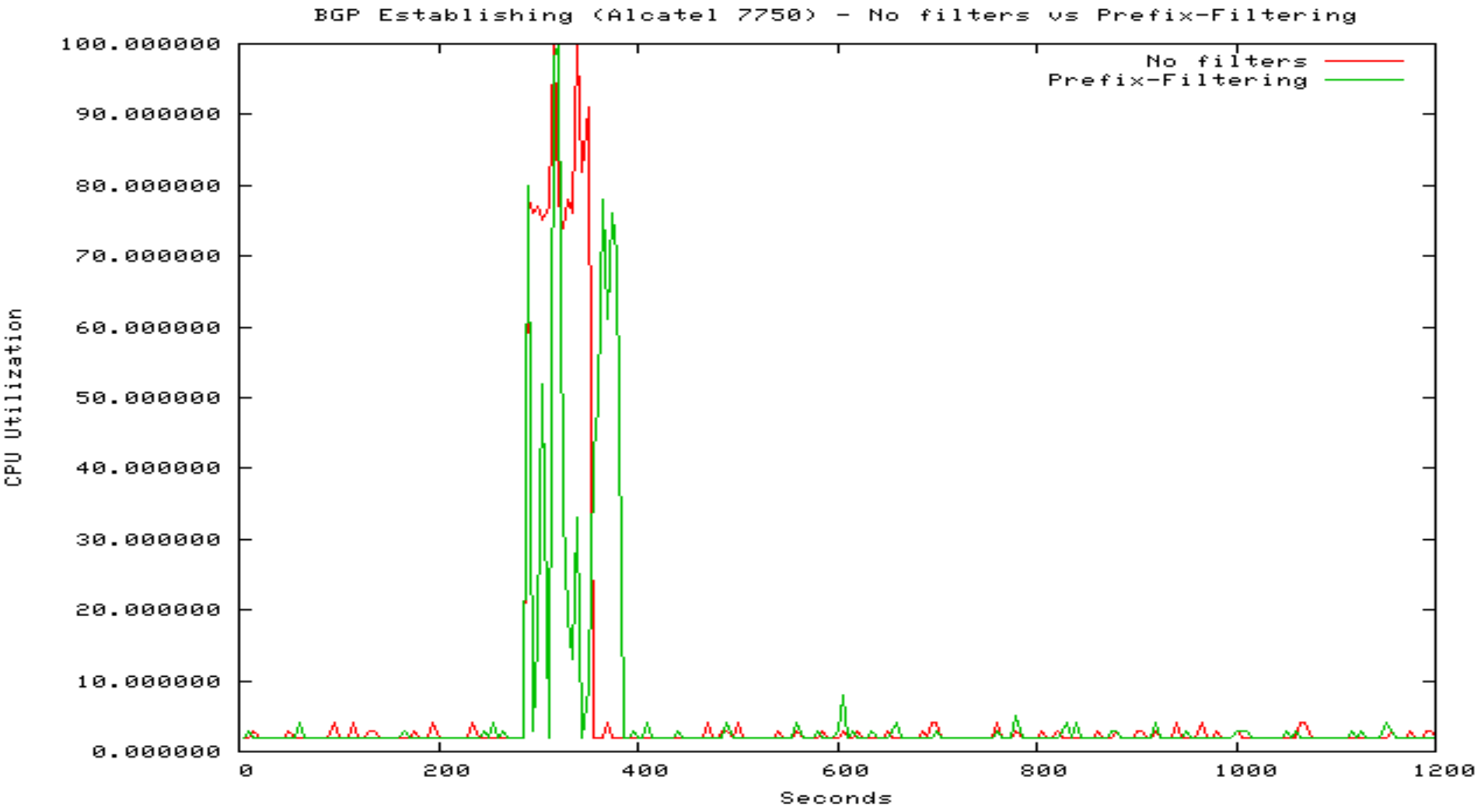
Testing Results – Cisco - Sup720



Testing Results - Alcatel

- Configuration size:
  - Typical configuration upload time: ~27 seconds
  - Prefix-List configuration was ~3.2MB
  - 114 Seconds to insert & commit configuration

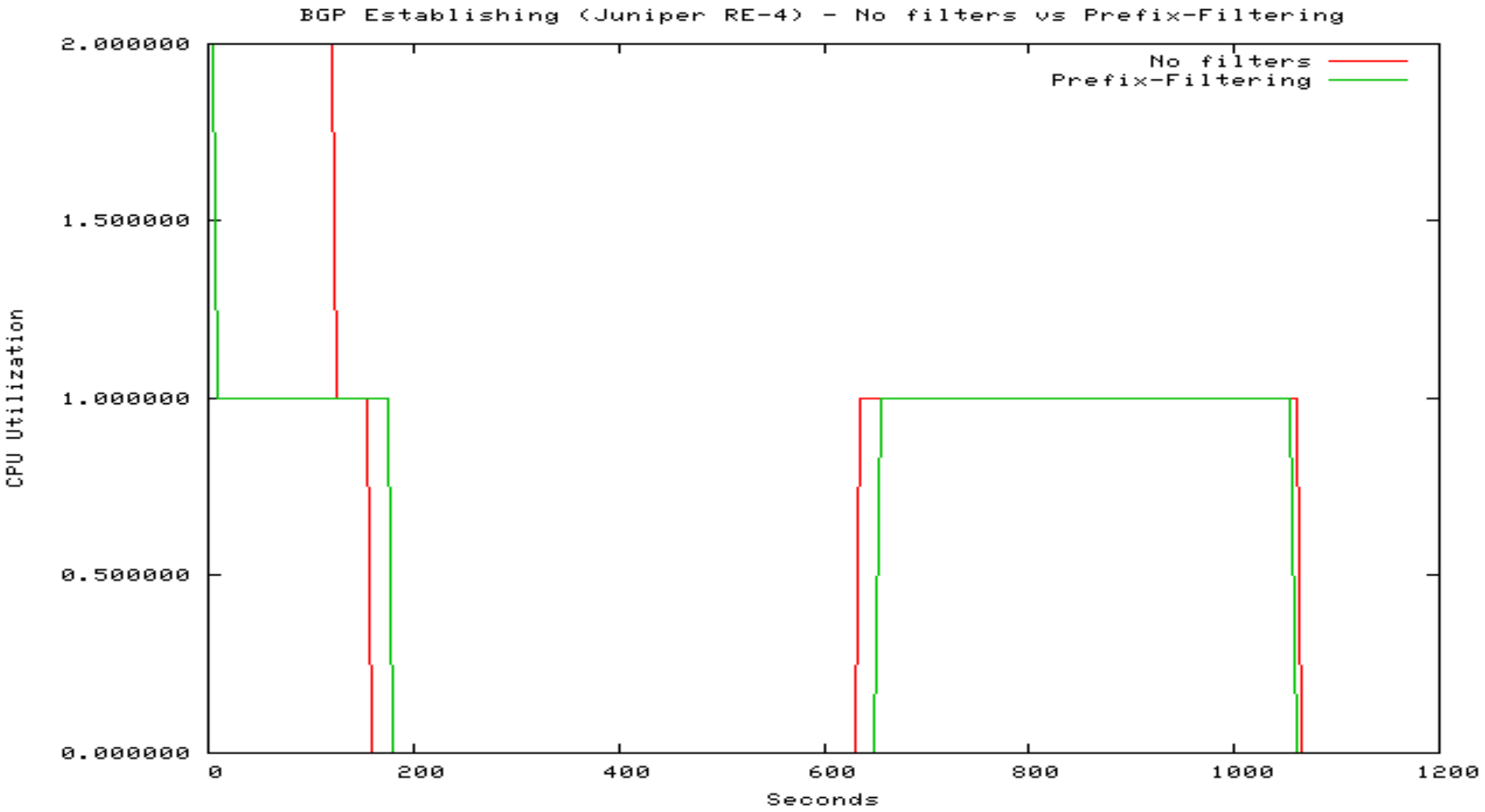
Testing Results – Alcatel 7750



Testing Results – Juniper

- Configuration size:
  - Similar to Cisco
  - Time to insert/commit not measured
- Overall
  - Almost no measurable impact on CPU at all

Testing Results – Juniper



•Testing Conclusions

- Enabling per-peer prefix-filters does not have any major impact on router operations today (as far as CPU utilization is being used to gauge it).
- “Flat-File” router configurations may not be the best method to upload prefix-lists. Perhaps uploading a prefix-list to its own file and call that prefix-list in the router configuration.

•Conclusions

- Something is needed to drive adoption of IRRs
- Possible to validate peer routes as advertised by large networks even with incomplete IRRs
- Possible to manage/deploy very large configs on most modern routers (with exceptions)
- Large operators who already filter all customer sessions should consider filtering peers using these (or similar) techniques.

Thank You

Jim Deleskie	**jim.deleskie@teleglobe.ca**
Alin Popescu	**alin@renesys.com**
Tom Scholl	**ts3127@sbc.com**
Todd Underwood	**todd@renesys.com**

Additional Slides

Router Configurations

- Cisco GSR – GRP-B, PRP-1, PRP-2
- Alcatel 7750SR
- Juniper config available from authors on request

Router Configurations (Cisco)

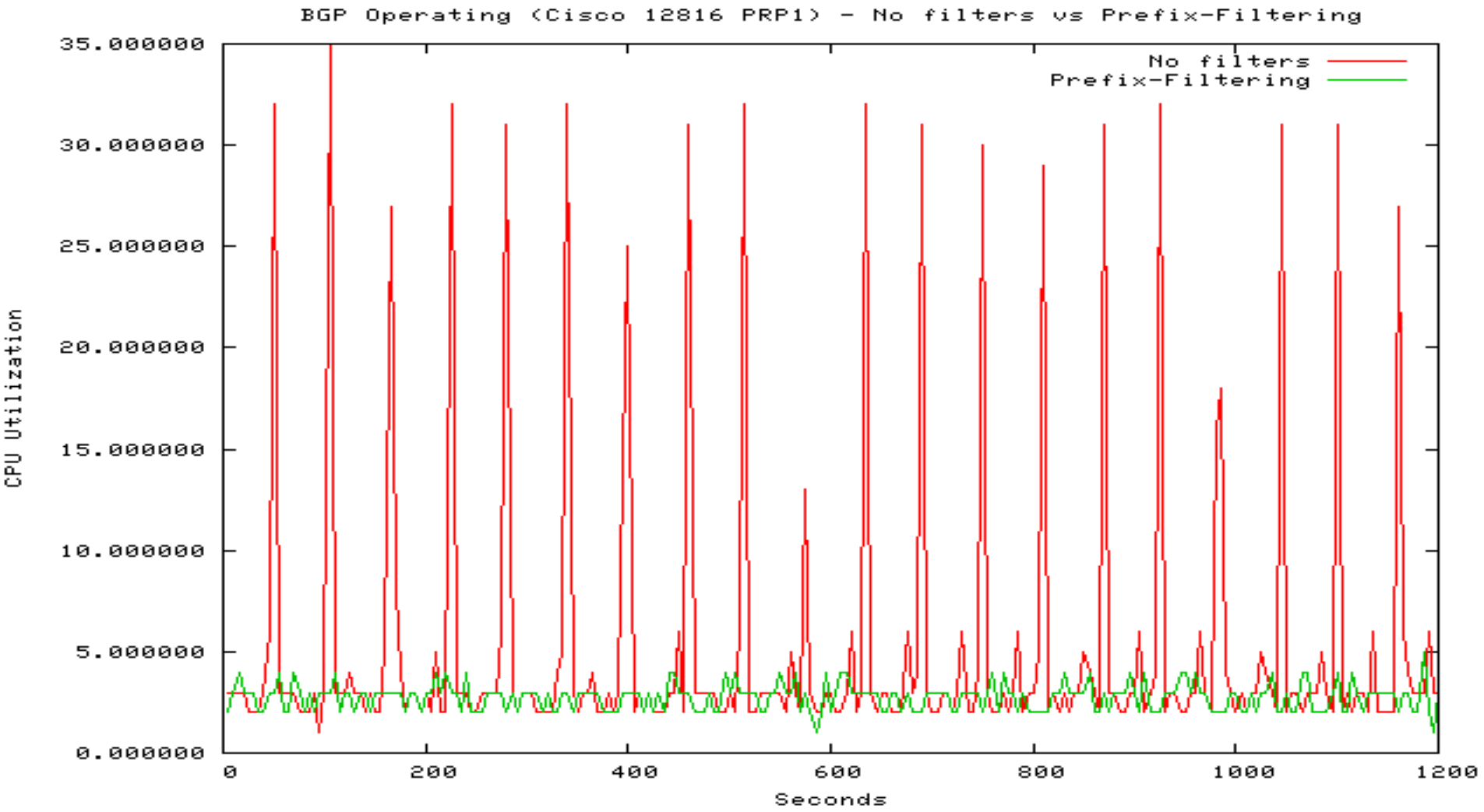
Cisco Configuration:

```
 neighbor PEER peer-group
 neighbor PEER update-source Loopback0
 neighbor PEER next-hop-self
 neighbor PEER soft-reconfiguration inbound
 neighbor PEER route-map PEER-IN in
 neighbor PEER route-map PEER-OUT out

 neighbor x.x.x.x remote-as <Remote ASN>
 neighbor x.x.x.x peer-group PEER
 neighbor x.x.x.x ebgp-multihop 255
 neighbor x.x.x.x prefix-list <ASxxxx> in

route-map PEER-IN permit 10
 set metric 0
 set community 123:123

route-map PEER-OUT deny 5
 match ip address prefix-list BOGONS
!
route-map PEER-OUT permit 10
 match community INTERNAL CUSTOMER
 set metric 0
 set community none

ip prefix-list ASxxxx permit y.y.y.y/yy
```

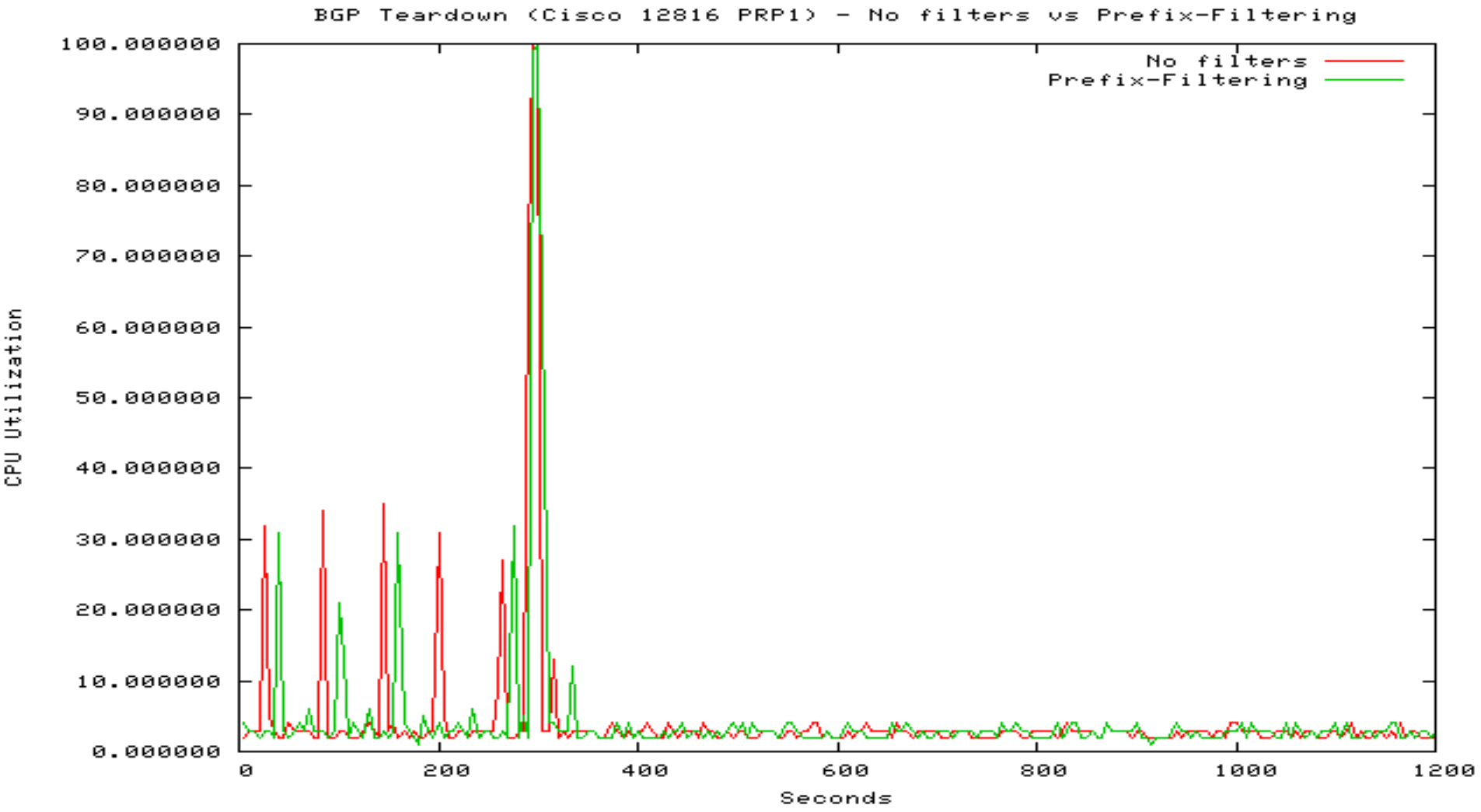
Router Configurations (Alcatel)

```
bgp
    group "PEER"
        local-address z.z.z.z
        import "PEER-IN"
        export "PEER-OUT"
        local-as <LocalASN>
        neighbor x.x.x.x
            multihop 255
            type external
            import "<Remote ASN>-IN"
            peer-as <Remote ASN>
        exit

    policy-statement "<ASN>-IN"
        entry 1
            from
                prefix-list "<ASN>"
            exit
            action accept
            exit
        exit
        default-action reject
    exit

    policy-statement "PEER-OUT"
        entry 10
            from
                prefix-list "BOGONS"
            exit
            action reject
        exit
        entry 20
            from
                protocol bgp
                community "INTERNAL"
            exit
            action accept
                metric set 0
            exit
        exit
        default-action reject
    exit
    policy-statement "PEER-IN"
        entry 10
            from
                prefix-list "BOGONS"
            exit
            action reject
        exit
        entry 20
            action accept
                community replace "PEER"
            exit
        exit
    exit

    prefix-list "ASN"
            prefix y.y.y.y/yy exact
    exit
```

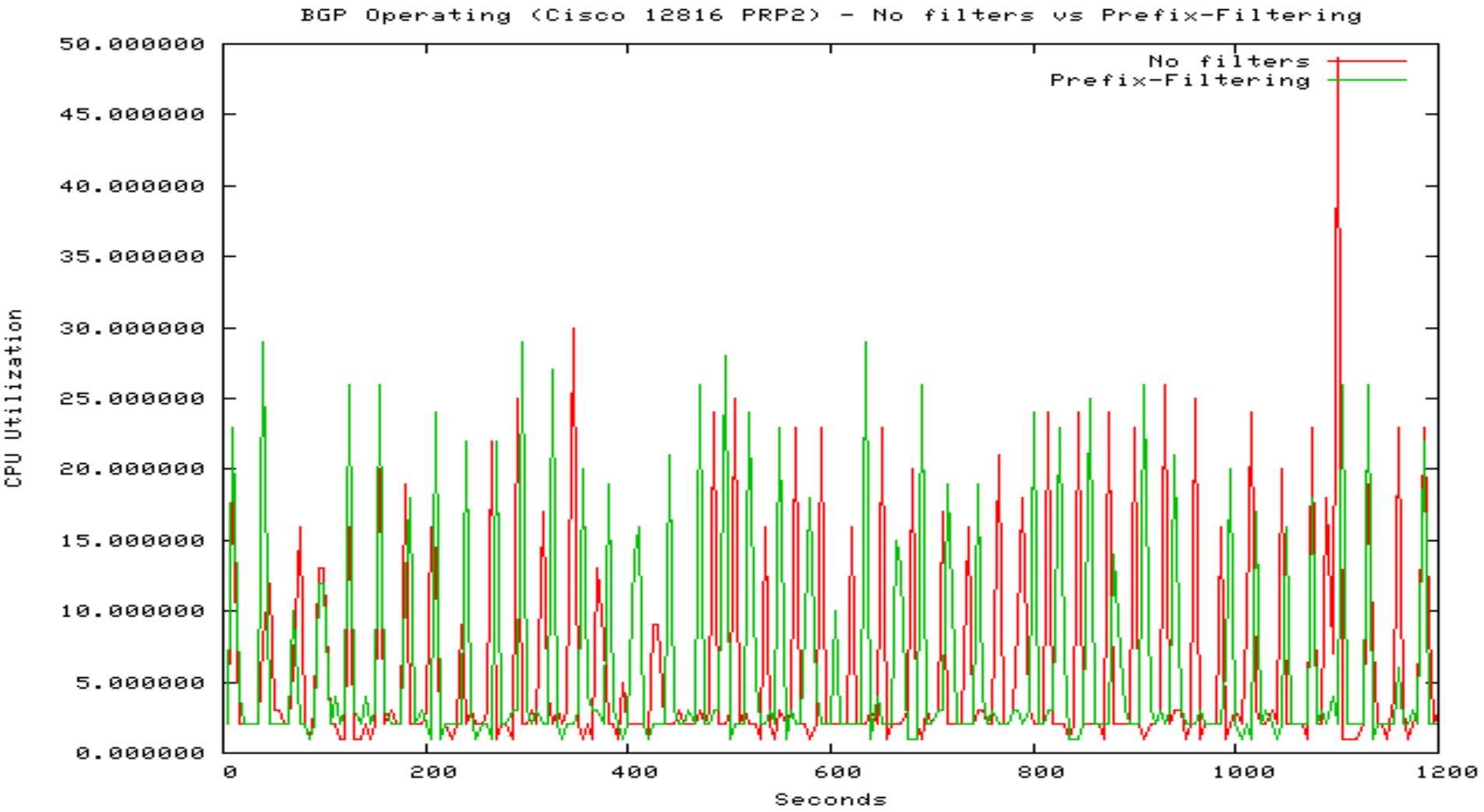
Testing Results – Cisco - PRP1



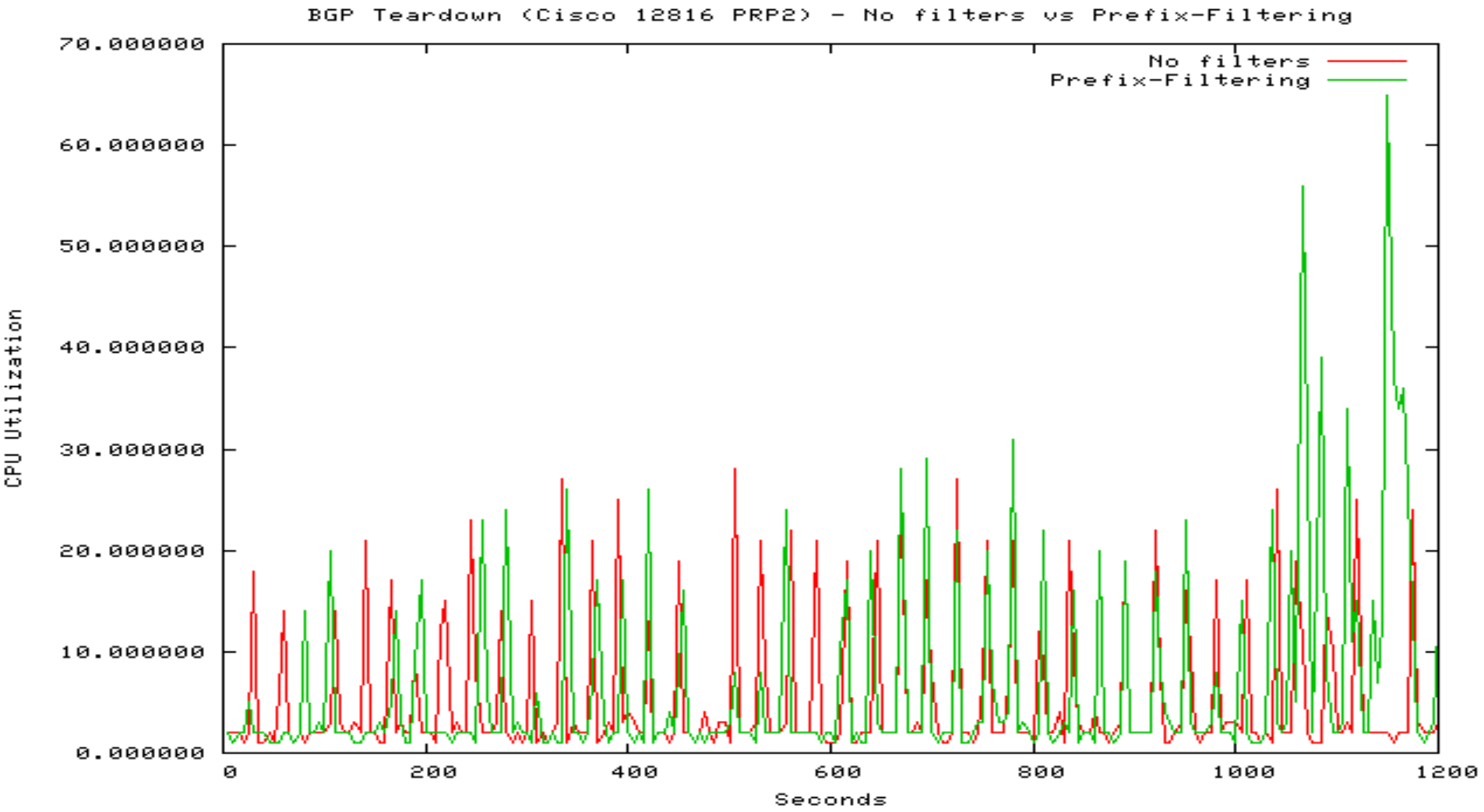
Testing Results – Cisco - PRP1



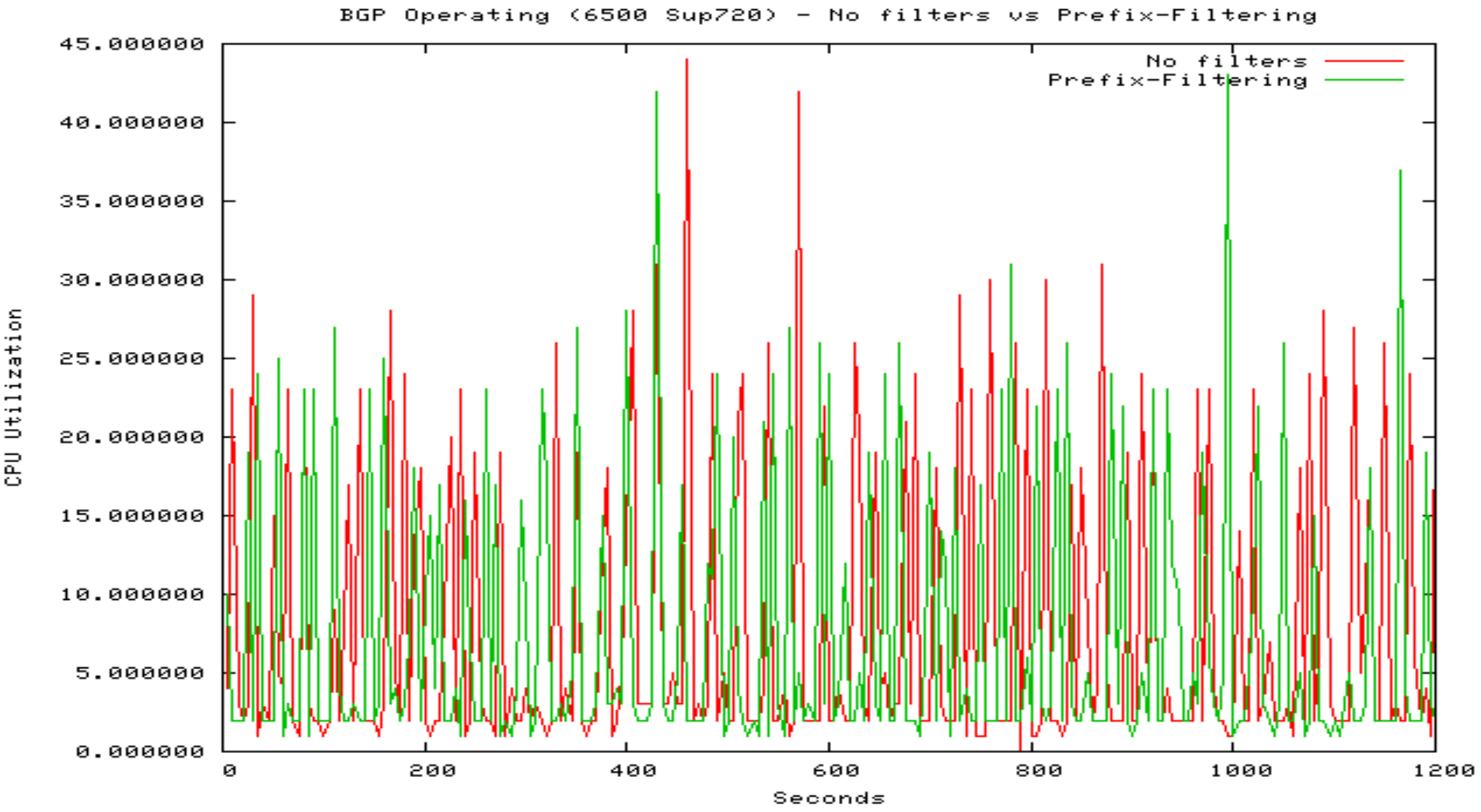
Testing Results – Cisco - PRP2



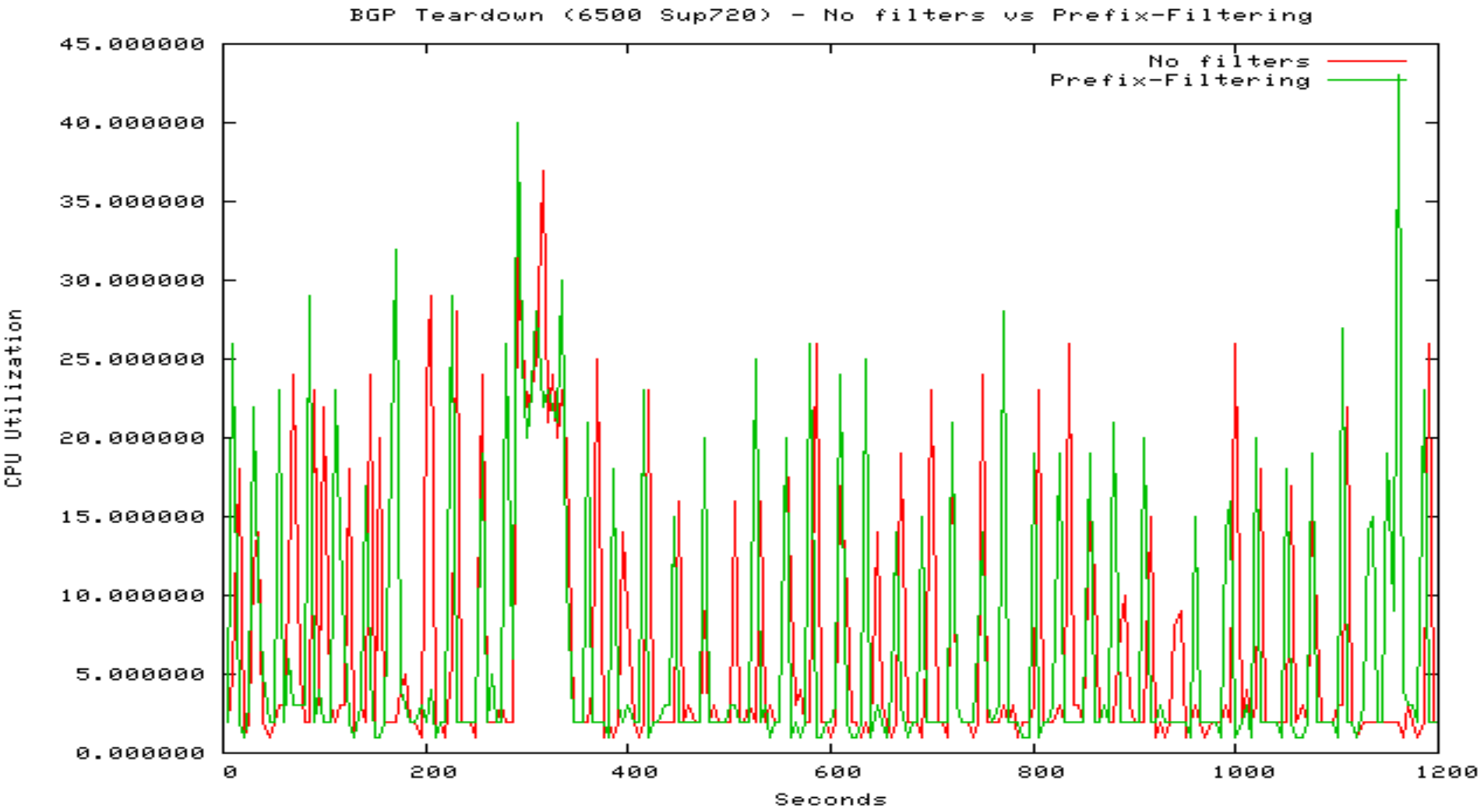
Testing Results – Cisco - PRP2



Testing Results – Cisco - Sup720

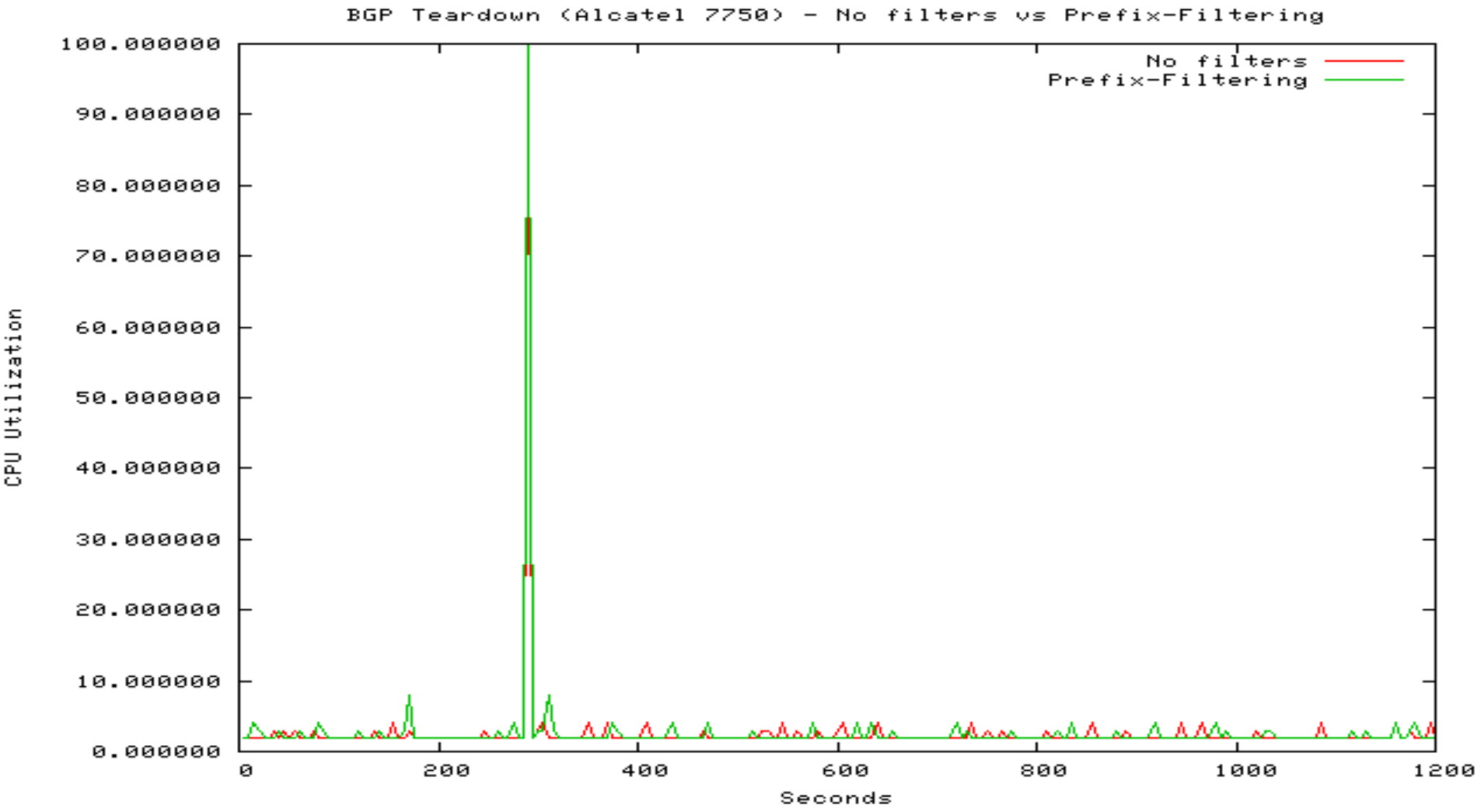


Testing Results – Cisco - Sup720

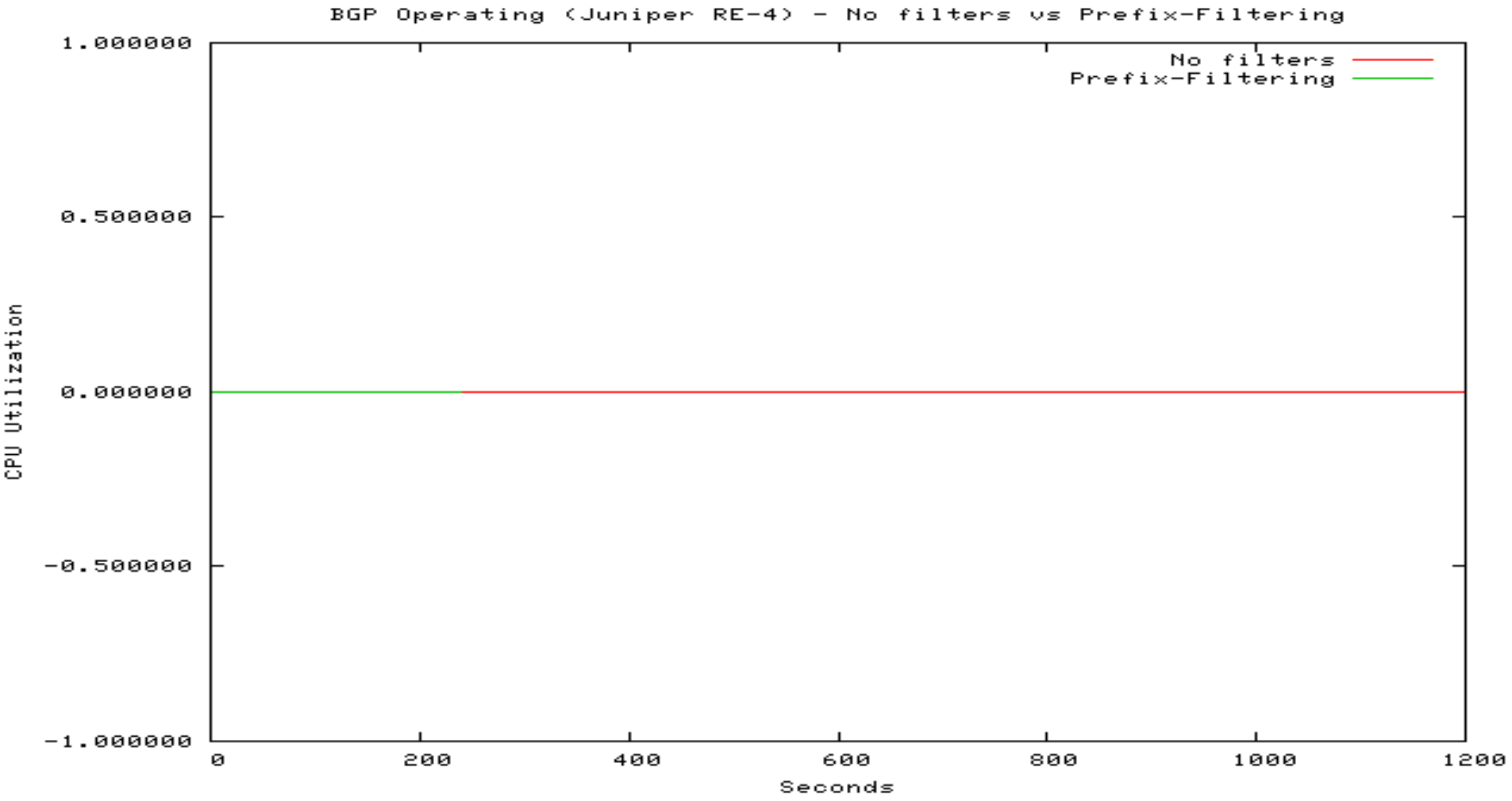


Testing Results – Alcatel 7750

Testing Results – Alcatel 7750



Testing Results – Juniper RE-4.0



Testing Results – Juniper - RE-4.0