A Technique for Counting NATted Hosts

Steven M. Bellovin

smb@research.att.com

AT&T Labs Research

Presented by **Matthew Packard**

July 4, 2003

About the Author - Steven Bellovin

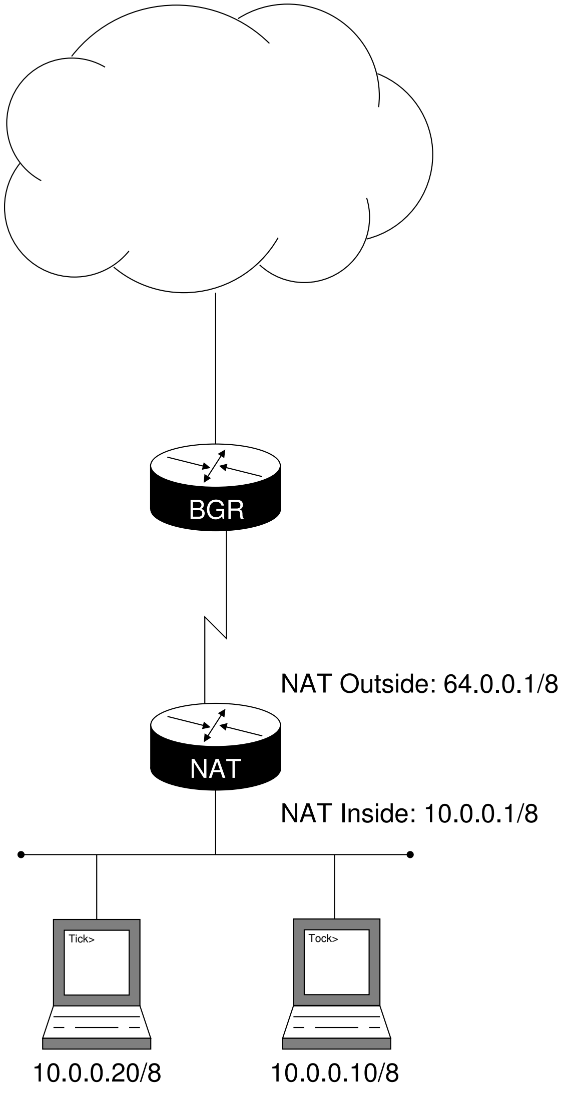
Steven Bellovin → http://www.research.att.com/∼smb

- AT&T Fellow in Network Services Research Lab
- Adjunct professor of Computer Science, University of Pennsylvania

- Academic history:
  - BA from Columbia University
  - MS in Computer Science from UNC, Chapel Hill
  - PhD in Computer Science from UNC, Chapel Hill
- Research interests:
  - networks
  - security
  - why the two don't get along

NAT Overview - How Does It Work & Why Do We Need It?

- NAT gateway translates/rewrites addresses
  - interior (10/8) ↔ exterior (64/8)
- Separates interior hosts by source port
  - available gateway source ports limiting factor
- NAT useful for:
  - home user with restrictive service agreement
  - corporate branch user with few hosts
  - cost-effective load balancer (web)
  - cost-effective firewall
  - security conscious Internet user
    - must break gateway to gain internal access
  - address space migration/bridging



Typical NAT Design

NAT Overview (Continued) - Compatible Protocols

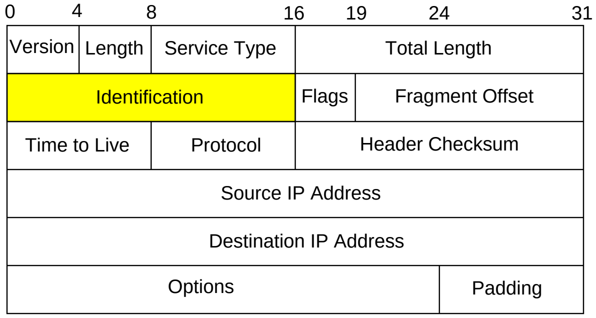
- NAT works with numerous TCP and UDP protocols
  - Easily translated - no data-embedded addresses
    - HTTP
    - TFTP
    - telnet
    - finger
    - NTP
    - NFS
  - Not so easily translated - data-embedded addresses
    - ICMP
    - FTP
    - NetBIOS (NetBT)
    - RealAudio
    - DNS
    - PPTP
    - H.323v2

Introduction - Necessity for NAT and NATted Host Counting

- Major reason(s) for using NAT:
  - lack of IPv4 addresses (primary)
  - security
  - rest of slide 3 uses

- Why pursue NAT counting in the first place?
  - accurate representation of what's on the Internet
  - evil ISPs who like to charge per host

- Major indicator of NATting - `IPid` field



0 4 8 16 19 24 31

Version	Length	Service Type	Total Length
Identification		Flags	Fragment Offset
Time to Live	Protocol	Header Checksum	
Source IP Address			
Destination IP Address			
Options			Padding

Introduction (Continued) - `IPid` Applicability & Issues

- Why use the `IPid` field?
  - generally implemented in most OSes as a global counter
  - different from TCP sequence number - unique per connection

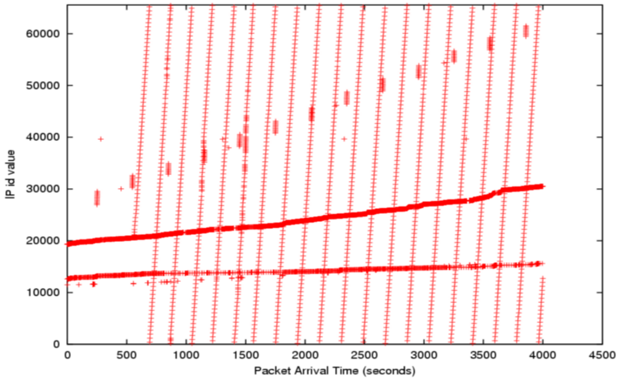
- Complicating factors using only the `IPid` field:
  - not all packets destined for Internet - gaps result
  - some OSes use byte-swapped `IPid` field - harder to see linear trend
  - Linux implements `IPid` only for fragment reassembly - set to 0 (MTU discovery)
  - Free/Net/OpenBSD use a randomized `IPid` value
  - Solaris divvies address space by ⟨ source, destination, protocol ⟩ triple

Introduction (Continued) - `IPid` Intended Uses & Assumptions

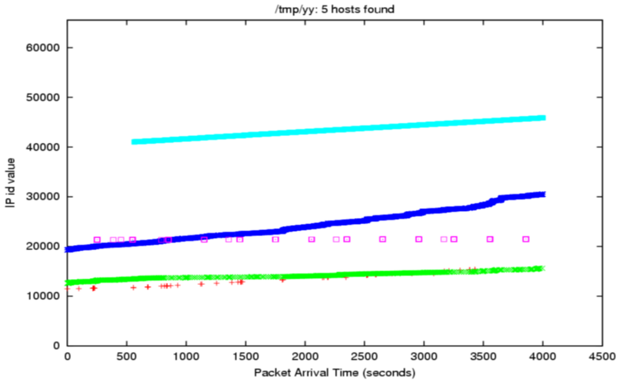
- `IPid` defined by RFC 791 to be just unique, not necessarily a counter
- `IPid` must be unique per ⟨ source, destination, protocol ⟩ triple
  - assists in packet defragmentation
- Transmission limited to ∼7.8 Mbps with 10 second packet lifetime (150 bytes)
- Transmission limited to ∼78 Mbps with 10 second packet lifetime (1500 bytes)

Algorithm - Implementation

- Upon receipt of new `IPid`, add to 'best' sequence:
  - `IPid` over *timelim* seconds → no match
  - `IPid` one higher than previous → *Perfect* match
  - `IPid` within *gaplim* of previous → *OutOfOrder*
  - `IPid` close but seen before → *Dup*



- Adjacent sequences coalesced (close enough)
  - `IPid`s within *gapfac* · *gaplim* or *timefac* · *timelim*



- Sequences less than *fsize* are discarded (bad guesses?)

- Packets with 0 `IPid` dropped (mod 2^{16} wrap)

- Both byte-swapped and normal counters checked
  - packet added to best match in either one
  - upon equal/no match → add to both

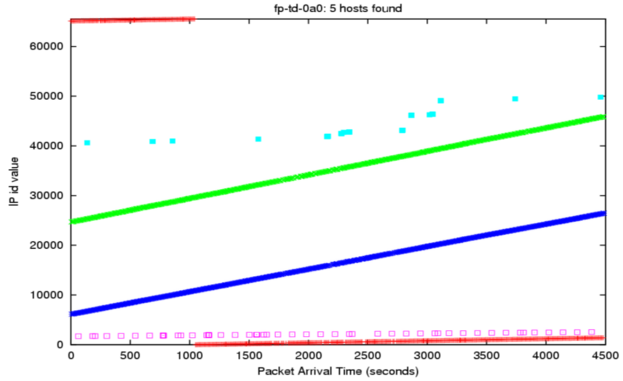
Parameter	Value
timelim	300
gaplim	64
timefac	5
gapfac	70
fsize	50

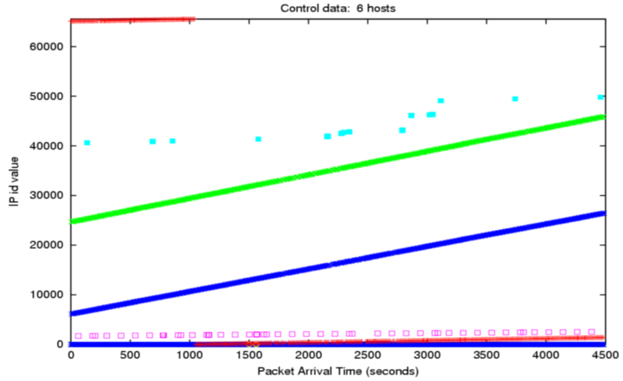
Observations & Limitations - Test Sources & Restrictions

- NAT data not derived from ISP end points
  - monitors need to be near provider termination point
  - miscounting due to routing - too easy to do so
- `IPids` culled from active client hosts
  - no examination of IP addresses - pseudo NAT design
- Data collected compared with actual IP addresses
  - not off by more than one - missed due to thresholds
- First 16 bytes of IP header used - IP destination stripped
  - ensures security - sending rate only known value
- Client only subnet monitored - servers change `IPids` too rapidly

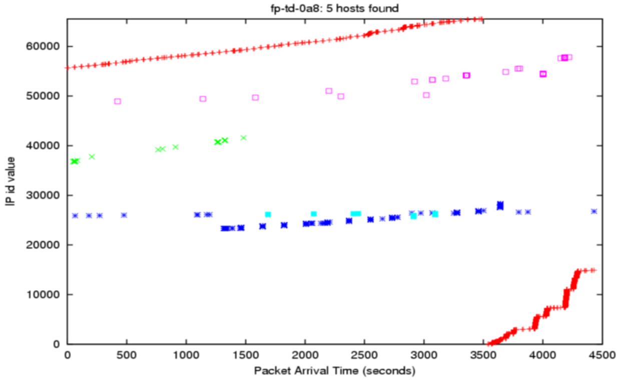
Observations & Limitations (Cont) - Results vs Actual Hosts

- Top → algorithm results
- Bottom → actual hosts
- Almost identical, except for:
  - `IPids` of 0 from OSPF router
  - missed one host - very low volume sender
    - `IPids` of 0 - 5 packets ($t = 1500$)

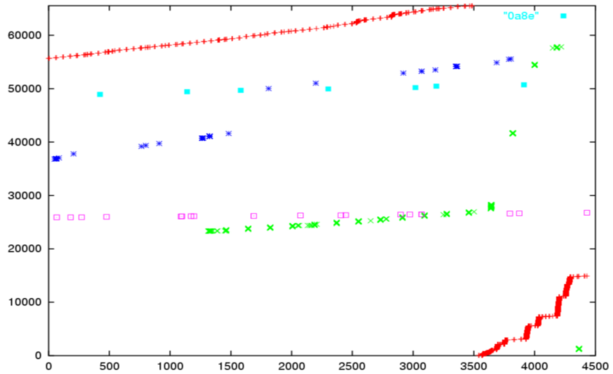




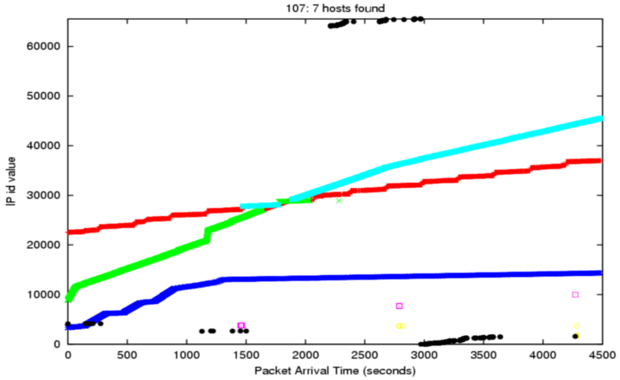
Observations & Limitations (Continued) - Collisions



- ❖ Top → algorithm results
- ❖ Middle → actual hosts



- ❖ `IPid` of 26000 and $t = 1300$, second host starts
  - ✥ assumed one sequence from $t = 1300$ to $t = 1600$
  - ✥ noticed second host at $t = 1600$
  - ✥ mismatched sequences until $t = 3750$
- ❖ Bottom → two hosts misinterpreted as three



- ❖ Large number of packets don't confuse analysis
  - ✥ gaps in the sequence IDs do - loopback, internal traffic
- ❖ Tool best used for small sites - home user, hotel

Privacy Issues - How To Obscure Your NATted Hosts

- Recall:
  - NAT gateway has one or a small number of IPs
  - uniqueness required per destination - TCP fragmentation
- Then how do we confuse or prevent these analysis methods?
  - rewrite all packets with *DF* (Don't Fragment) - leaks possible
  - rewrite all packets with NAT-centric unique `IPids`
  - upon fragment receipt, rewrite all fragments with same unique ID
    - maintain state per packet - high overhead
  - use a random number generator for the `IPid` - keyed (Free/Net/OpenBSD)

Behavior Of Commercial NAT Devices - Why They're Cheap

- SOHO devices tested, plus IPFilter (NAT portion)
- Tests included small and large (fragment-able) ICMP and TCP packets
  - *DF* bit set and not set
- Results show *no* rewriting of the `IPid`
- Next version: purposely duplicate `IPids` to hide hosts?

Future Work - Room For Improvement

- Better sequence detection - perhaps using image processing instead?
- Utilize other header info besides `IPid`:
  - ⟨ source address, source port, destination address, destination port ⟩
- Other protocols - IPsec sequence numbers and RTP timestamps
- Passive fingerprinting - determine host types, not necessarily number of them
  - SYN and ICMP packet analysis:
    - IP → TOS (ECN), total length, IPid, TTL
    - TCP → source port, window, options (MSS, timestamp, wscale, SackOK, Nop)
  - http://www.incidents.org/papers/OSfingerprinting.php
- Other tools - sFlow → http://www.sflow.org/detectNAT

Conclusion - Quick Recap

- IPid used as a global connection counter - easy to fingerprint from
- Analysis of NATted hosts possible and quite accurate for small number of hosts
- Analysis falters under high node count, servers, or gapping
  - gaps prove to be fatal to the proper distinction of hosts
- Analysis thwarted by NAT gateway properly rewriting IPids

Discussion

- ❖ Questions?

References

- D. Hucaby, S. McQuerry, *Cisco Field Manual: Router Configuration*, Cisco Press, 2002, pp. 237-244.

- B. Conoboy, E. Fichtner, *IP Filter Based Firewalls HOWTO*, Internet resource (http://coombs.anu.edu.au/∼avalon/ip-filter.html), 2002.

IPFilter - How To Get & Configure It

- Source → http://coombs.anu.edu.au/∼avalon/ip-filter.html

- Notable features:
  - packet filter
  - NAT capability
  - UNIX based (Solaris, Free/Net/OpenBSD)

- General configuration ruleset parlance:
  - `pass in quick on hme0 proto tcp from any to 64.0.0.1/8 port = 22 keep state`

- NAT configuration ruleset parlance:
  - `map hme0 10.0.0.0/8 -> 64.0.0.1/8`
  - `map hme0 10.0.0.0/8 -> 0/8`
  - `map hme0 10.0.0.0/8 -> 0/8 portmap tcp/udp 20000:30000`
  - `map hme0 10.0.0.0/8 -> 0/8 tcp/udp auto`

Cisco IOS NAT Configuration

- General configuration ruleset parlance:

  -
    ```
    interface ethernet 0
      ip address 10.0.0.1 255.0.0.0
      ip nat inside
    interface ethernet 1
      ip address 64.0.0.1 255.0.0.0
      ip nat outside
    ip nat inside source static tcp 10.0.0.10 22 64.0.0.1 22
    ip nat inside source static network 10.0.0.20 64.0.0.1 255.0.0.0
    ip nat pool inside1 10.0.0.0 10.255.255.255 netmask 255.0.0.0
    ip nat inside source list 101 pool inside1
    ip nat inside source route-map map1 pool inside1
    access-list 101 permit ip 10.0.0.0 0.255.255.255 any
    route-map map1 permit 10
      match ip address 101
    ```

Slide Generation Utilities

- The GIMP → http://www.gimp.org
  - PNG cropping/chopping
- ImageMagick → http://www.imagemagick.org
  - `convert` utility for PDF image extraction and PNG conversion
- LaTeX → http://www.tug.org
  - `pdflatex` utility for PDF slide output
- XFig → http://www.xfig.org
  - PDF/PS graphics creation utility
- Slide Generation Process:
  - scale original PDF to at least 4 times normal size:
    - `convert -density 300 -enhance -antialias nat.pdf nat.png`
    - `convert -blur 1x1 -crop <x>x<y>+<x>+<y> nat.png.1 nat.png.new`