An Untold Story of Middleboxs in Cellular Networks

Zhaoguang Wang, Zhiyun Qian, Qiang Xu, Z. Morley Mao, Ming Zhang

Sigcomm 2011

Presented by: Matthew Tan Creti

Overview

1. Introduction to NetPiculet
2. NATs in Cellular Networks
3. Firewalls in Cellular Networks
4. Conclusion

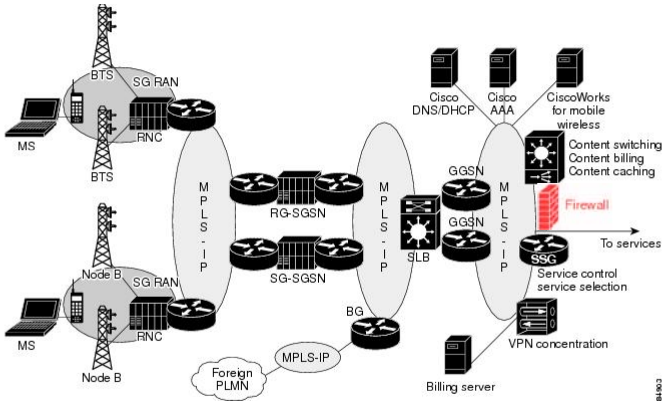
Motivation

- Cellular provider’s network policies are designed to fairly share limited resources and provide security
- These policies are mostly opaque to users, however, they directly impact the performance, energy, and security
- This work seeks to use measurements of cellular networks to infer cellular provider’s policies

Definition

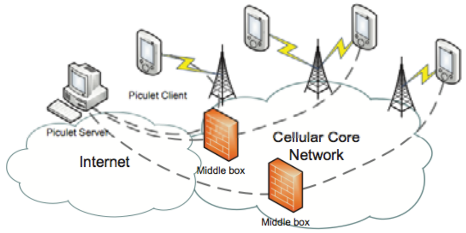
- *Middlebox*: a networking device that transforms, inspects, filters, or otherwise manipulates traffic for purposes other than packet forwarding
- Examples: NAT, firewall, IDS

Cellular Infrastructure



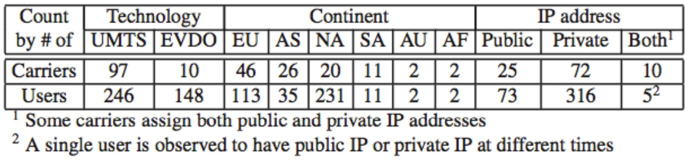
- Middleboxes are deployed near the GGSN (Gateway GPRS Support Nodes)

NetPiculet System



- NetPiculet runs on client mobile devices and the Piculet server
- Server’s upstream provider has no restrictive policies that interfere with experiments
- Clint runs tests in parallel, which finish in 10s
- Except TCP timeout test, which runs as background service

Carriers and Users Sampled



Count	Technology		Continent						IP address		
by # of	UMTS	EVDO	EU	AS	NA	SA	AU	AF	Public	Private	Both[1]
Carriers	97	10	46	26	20	11	2	2	25	72	10
Users	246	148	113	35	231	11	2	2	73	316	5[2]

[1] Some carriers assign both public and private IP addresses

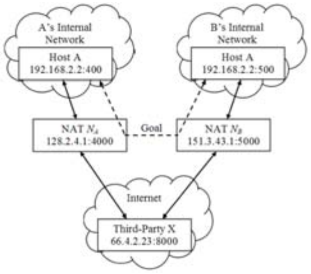
[2] A single user is observed to have public IP or private IP at different times

- Client software available on Android Market
- Attracted users by provided useful network information (e.g., will this P2P app run on this network)
- 393 unique users revealed information on 107 carriers

Overview

1. Introduction to NetPiculet
2. NATs in Cellular Networks
3. Firewalls in Cellular Networks
4. Conclusion

NAT Traversal



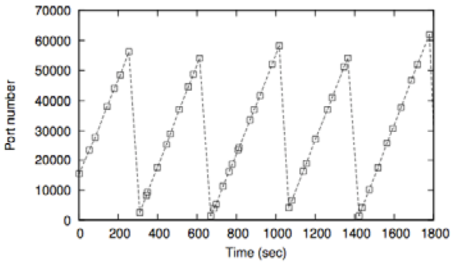
- NAT traversal is required by P2P applications
- Goal is to establish a TCP connection between A and B
- Many hacks exist, dependent mostly on what mapping method each NAT is using
  - When does the NAT assign a new external endpoint (e.g., per source or per connection)?
  - How is the external endpoint port number chosen (e.g., incremental or random)?

NAT Mapping Results

NAT Mapping	# carriers
Independent	30
Address and Port$_1$	15
Connection$_R$	19
Connection$_T$	5
Address and Port$_T$ & Connection$_T$	3
Total	72

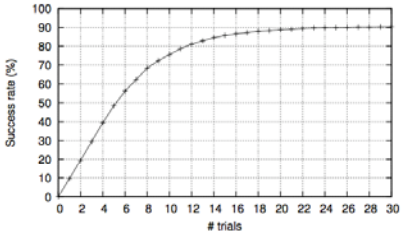
- NAT Mapping methods
  - Independent: external endpoint remains same for all connections
  - Address and Port: external endpoint changes when destination endpoint changes
  - Connection: external endpoint changes for each new connection
- Meaning of subscripts
  - 1: external port is incremented by 1
  - R: external port is random
  - T: described on next slide…

Time-dependent NAT mapping

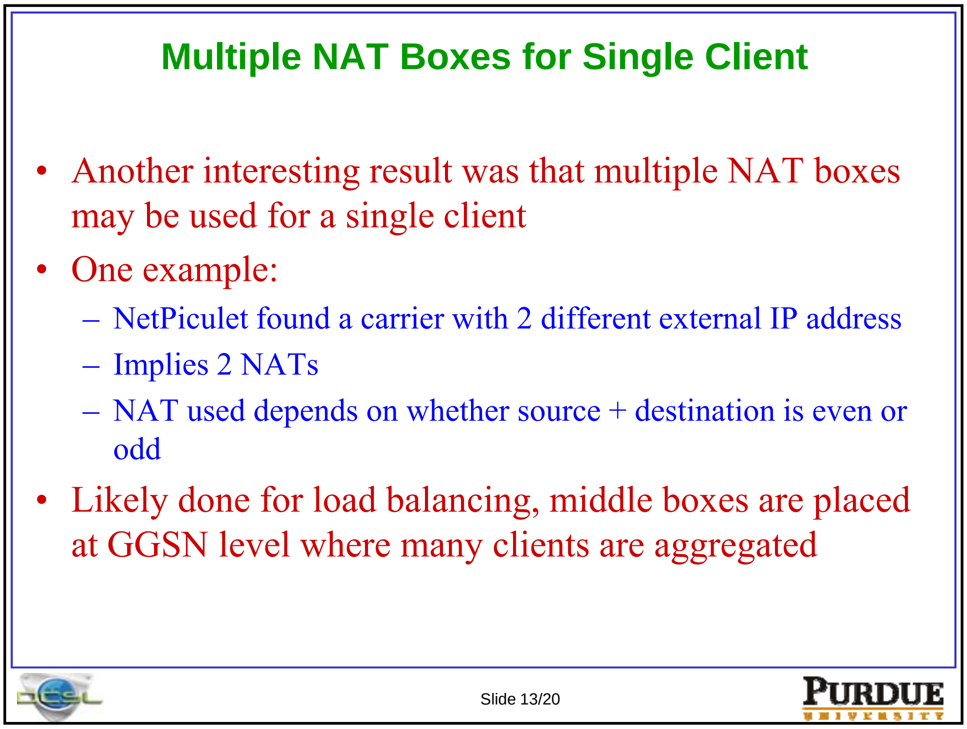


- 8 carriers where initially classified as Connection$_R$ or Address and Port$_1$
- Closer inspection showed they were time dependent
- This type of NAT has not been encountered in NAT traversal literature

Traversing Time-Dependent NAT

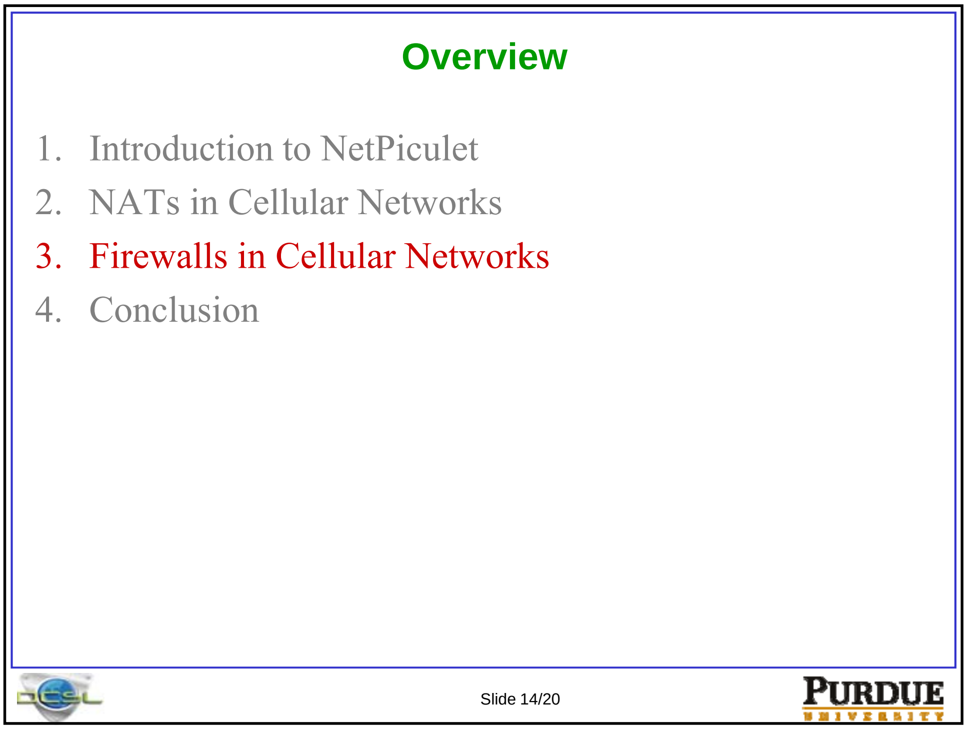


- State-of-the-art for random endpoint mapping
  - NATBlaster has A send 439 SYN packets
  - B sends SYN+ACK packets to random NAT$_A$ ports
  - Birthday paradox gives B a 95% chance of succeeding by its 440th attempt
- However, if we know mapping is time dependent we can use lighter weight approach
- Client B makes guesses of NAT$_A$ endpoint port in range [B_S+δ-n, B_S+δ+n]
  - B_S = b's external port discovered by server S
  - δ = port number increase (predicated by server S)
  - n = 15



Multiple NAT Boxes for Single Client

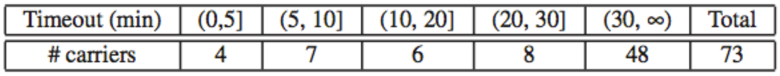
- Another interesting result was that multiple NAT boxes may be used for a single client
- One example:
  - NetPiculet found a carrier with 2 different external IP address
  - Implies 2 NATs
  - NAT used depends on whether source + destination is even or odd
- Likely done for load balancing, middle boxes are placed at GGSN level where many clients are aggregated



Overview

1. Introduction to NetPiculet
2. NATs in Cellular Networks
3. Firewalls in Cellular Networks
4. Conclusion

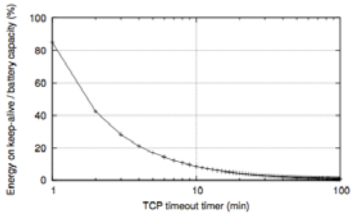
Testing TCP Connection Timeout



Timeout (min)	(0,5]	(5, 10]	(10, 20]	(20, 30]	(30, ∞)	Total
# carriers	4	7	6	8	48	73

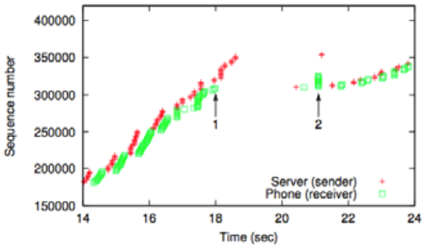
- NetPiculet opens multiple parallel connections without keep-alive option
- Each connection used to send message to server after specific amount of time
- 5, 10, 20, 30 minute idle time intervals tested

Energy Impact of TCP Connection Timeout



- Example:
  - MSN Talk needs to keep TCP connection open for long time
  - One major carrier had timeout of 255 seconds
  - MSN Talk was forced to re-establish connection, more delay and energy cost than sending keep-alive message
- Found 17% of battery capacity spent on keep-alive messages over one day for timeouts of less than 5 minutes
- There is trade off between client energy and provider’s firewall capacity
- One solution is push service framework

Evidence of Packet Buffering



- Major US carrier was found to buffer packets for over 1 hour
- Packet buffering at the firewall may be used for deep-packet inspection
- Prevents TCP fast retransmission
- In figure
  - Server packet lost at time 1
  - Server keeps sending to fill congestion window
  - Phone never sends duplicate acks that would normally trigger fast retransmission
  - Eventually server times out and retransmits the lost packet, at which time (2) the firewall releases all of the buffered packets

Impact of Packet Buffering

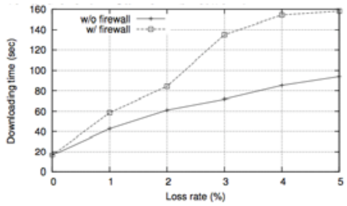


Figure 8: The average downloading time for 1MB file under different loss rates.

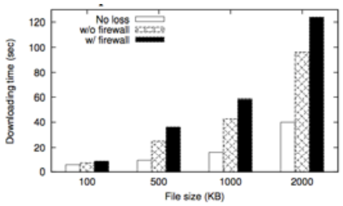


Figure 9: The firewall impact on downloading time for different file size under 1% loss rate.

- Packet buffering is more costly in cellular network because loss rates can be higher than in wireline networks
- Figure 8 shows that buffering increase download time of 1 MB file 50% for a loss rate of just 1%
- Figure 9 shows less impact (only 22% increase) for small 100KB files
- Recent study points out that TCP-based streaming applications that send large amounts of data contribute to majority of smartphone traffic
- Cellular radio stays in high power state during entire download process

Other Firewall Study Findings

- 4 of 60 cellular networks allow IP spoofing, which can make hosts vulnerable to scanning and battery draining attacks even though they are behind the firewall and NAT
- 11 of 73 carriers set TCP timeout to less than 10 minutes, based on study 30 minutes is recommended
- TCP out-of-order buffering behavior in come firewalls is causing unexpected interaction with common TCP behavior defined in TCP specifications

Conclusion

- NetPiculet approach to collecting results: build a tool that users want to use and mine the data (contact HRPP first)
- Cellular network middleboxes impact performance, energy, and security of client applications
- Found unusual NAT and firewall configurations, cellular providers could implement changes to improve user experience