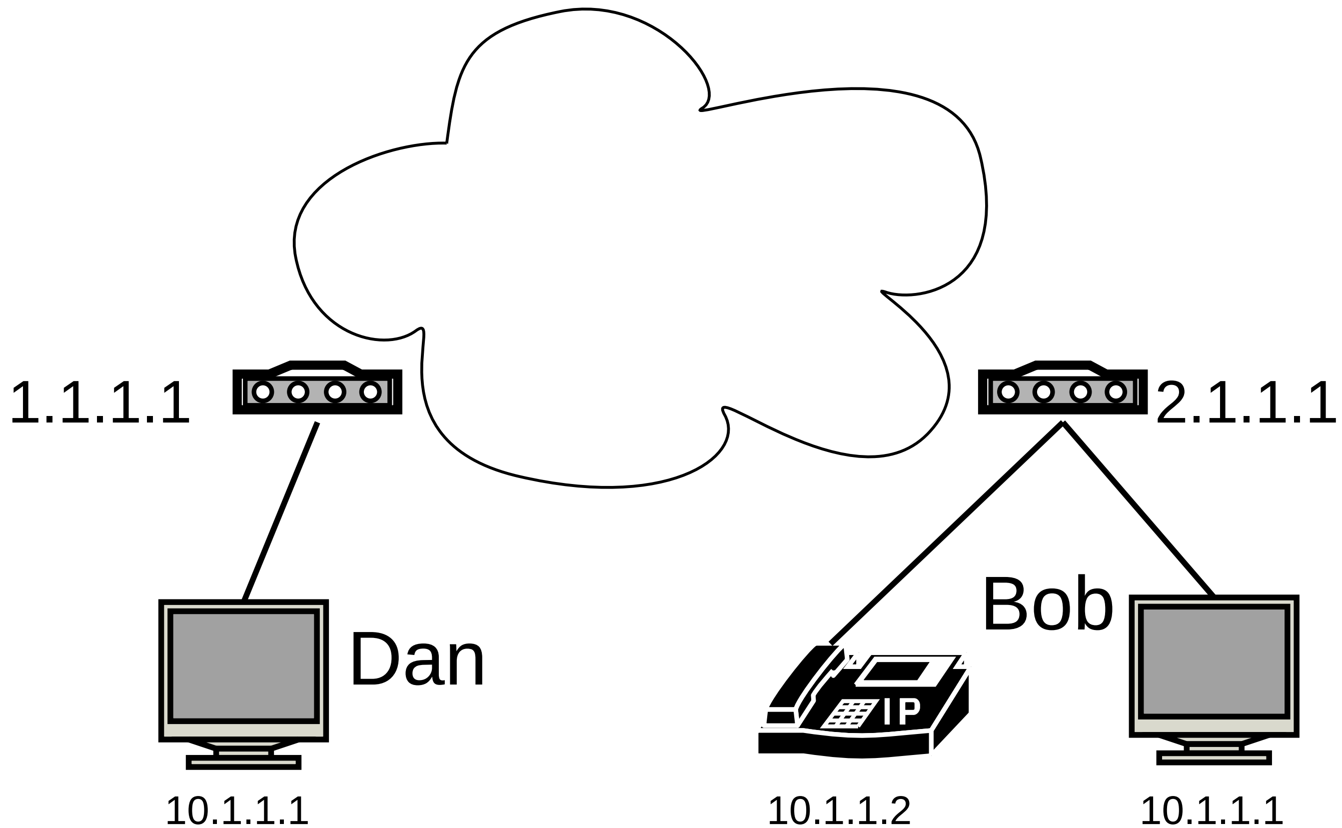
Characterization and Measurement of TCP Traversal Through NATs and Firewalls

Saikat Guha, Paul Francis

Cornell University

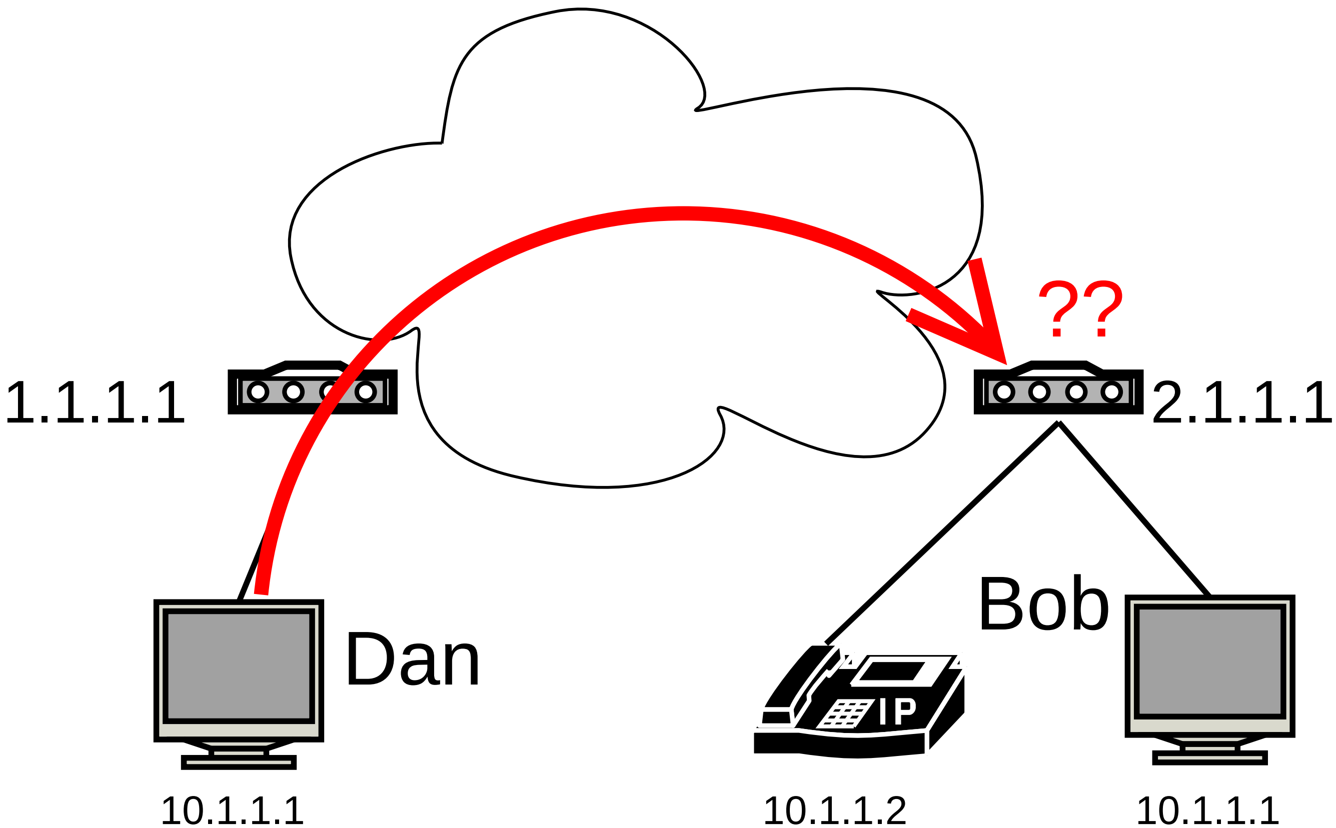
IMC 2005

P2P connectivity through NATs



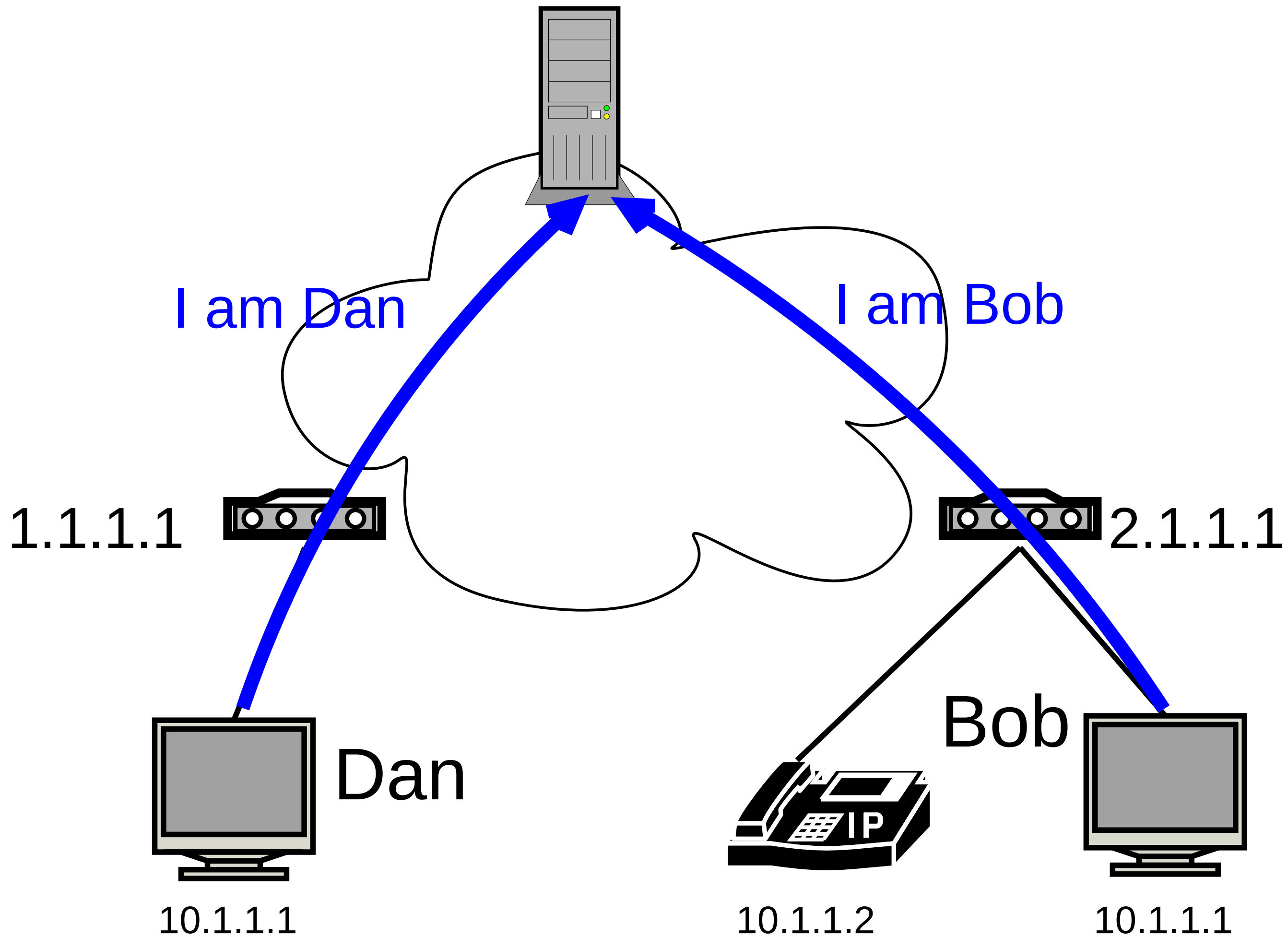
New inbound flows cannot be routed

P2P connectivity through NATs



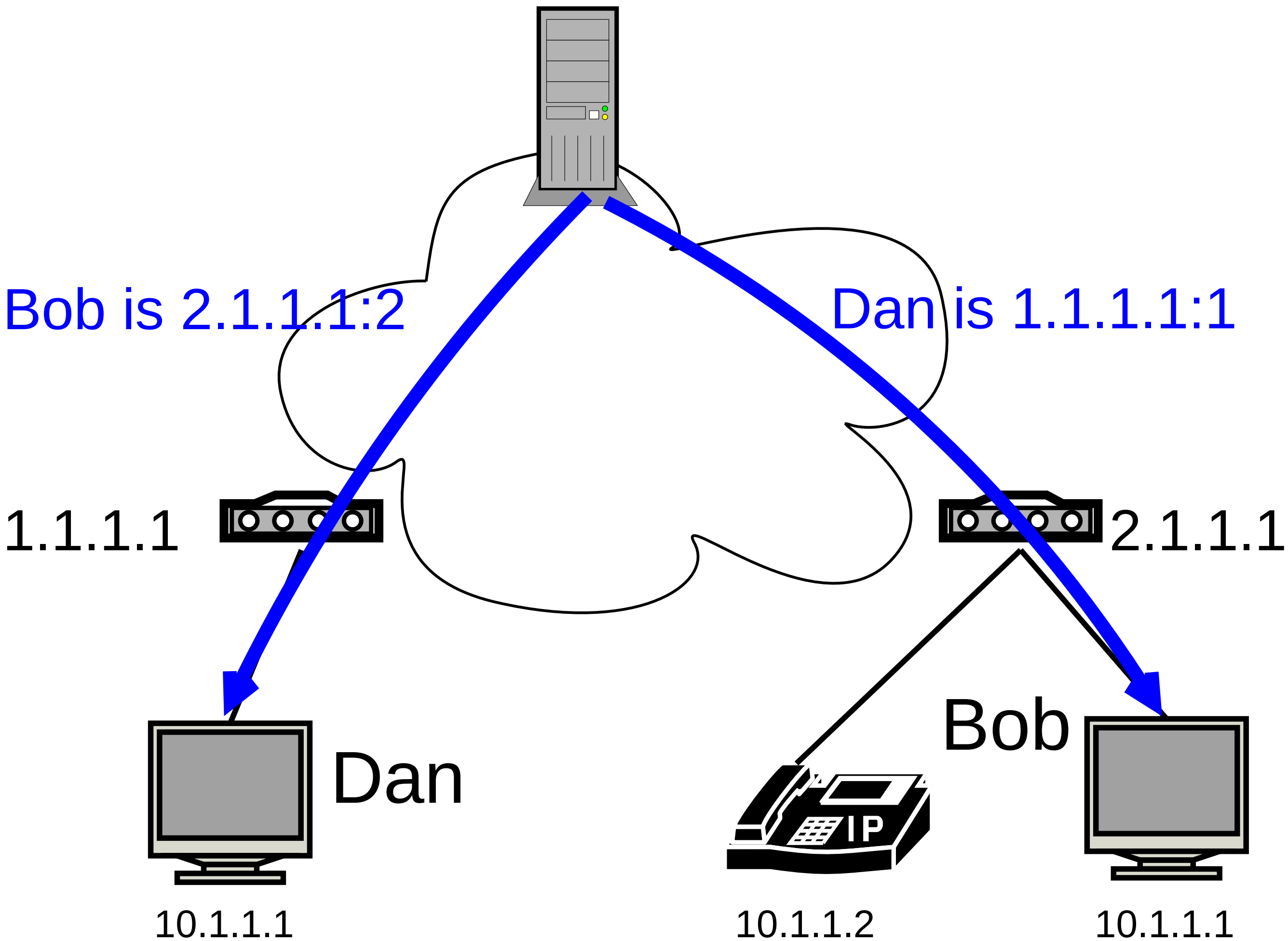
New inbound flows cannot be routed

P2P connectivity through NATs



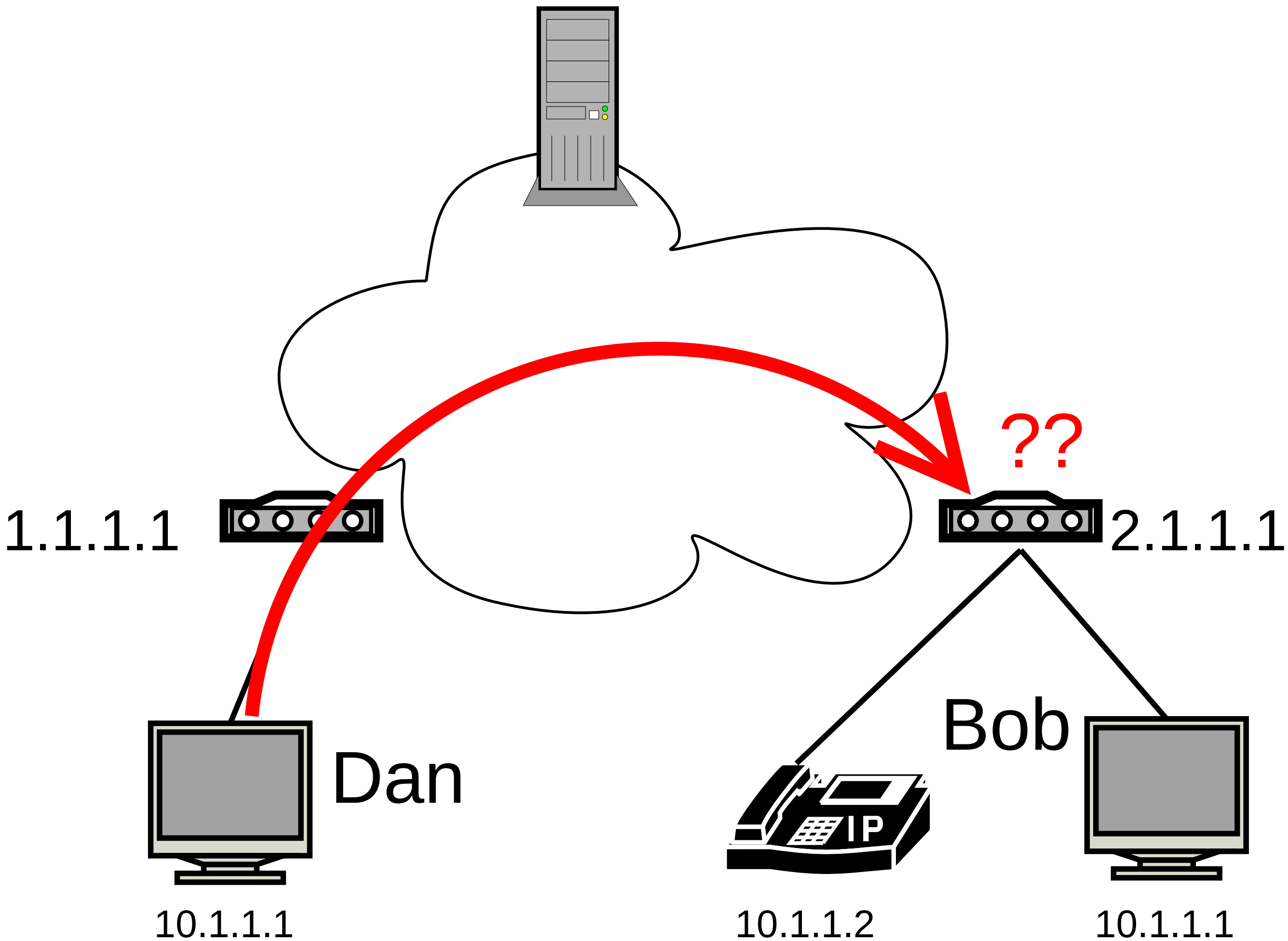
Basic solution for UDP

P2P connectivity through NATs



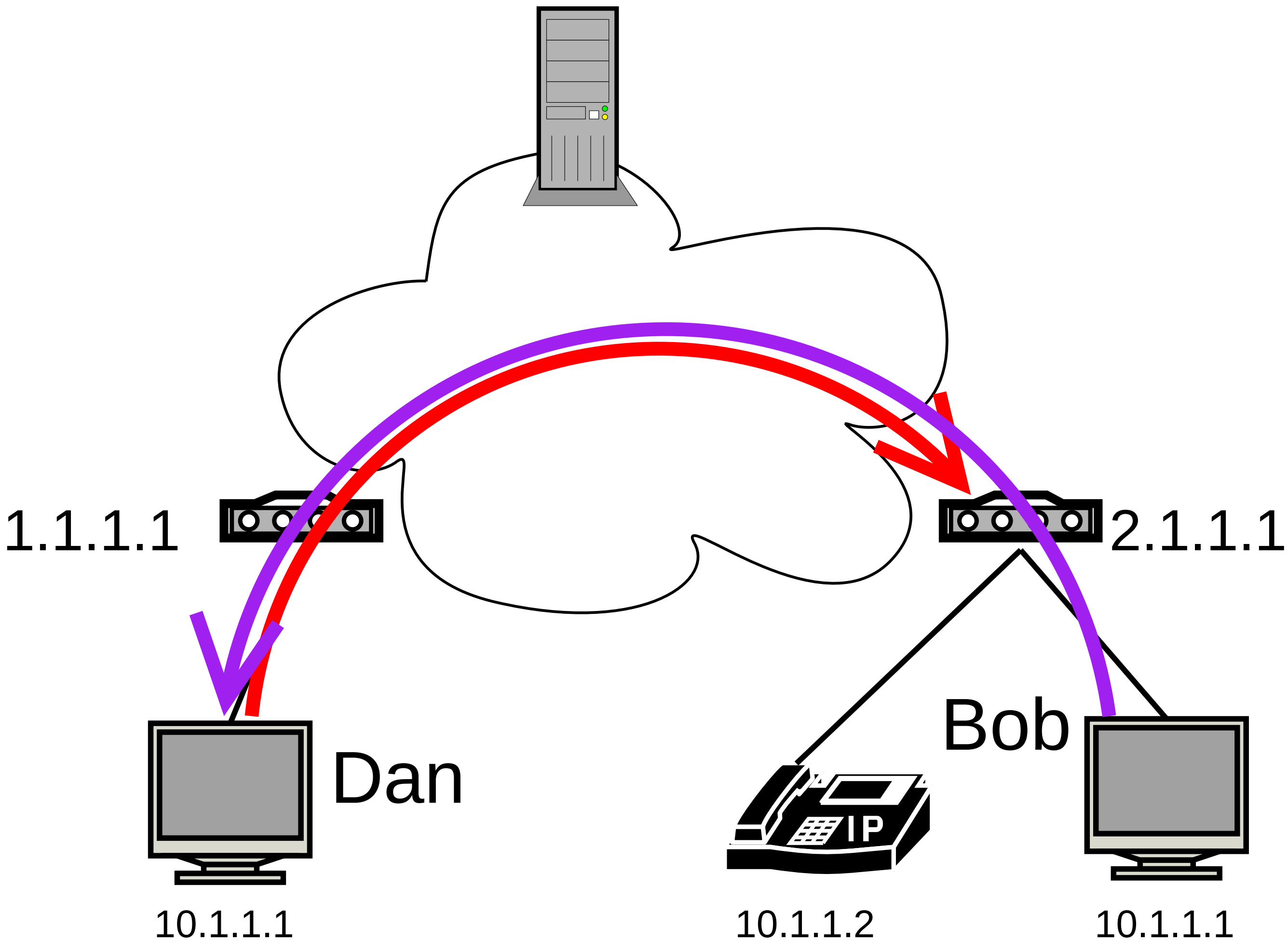
Basic solution for UDP

P2P connectivity through NATs



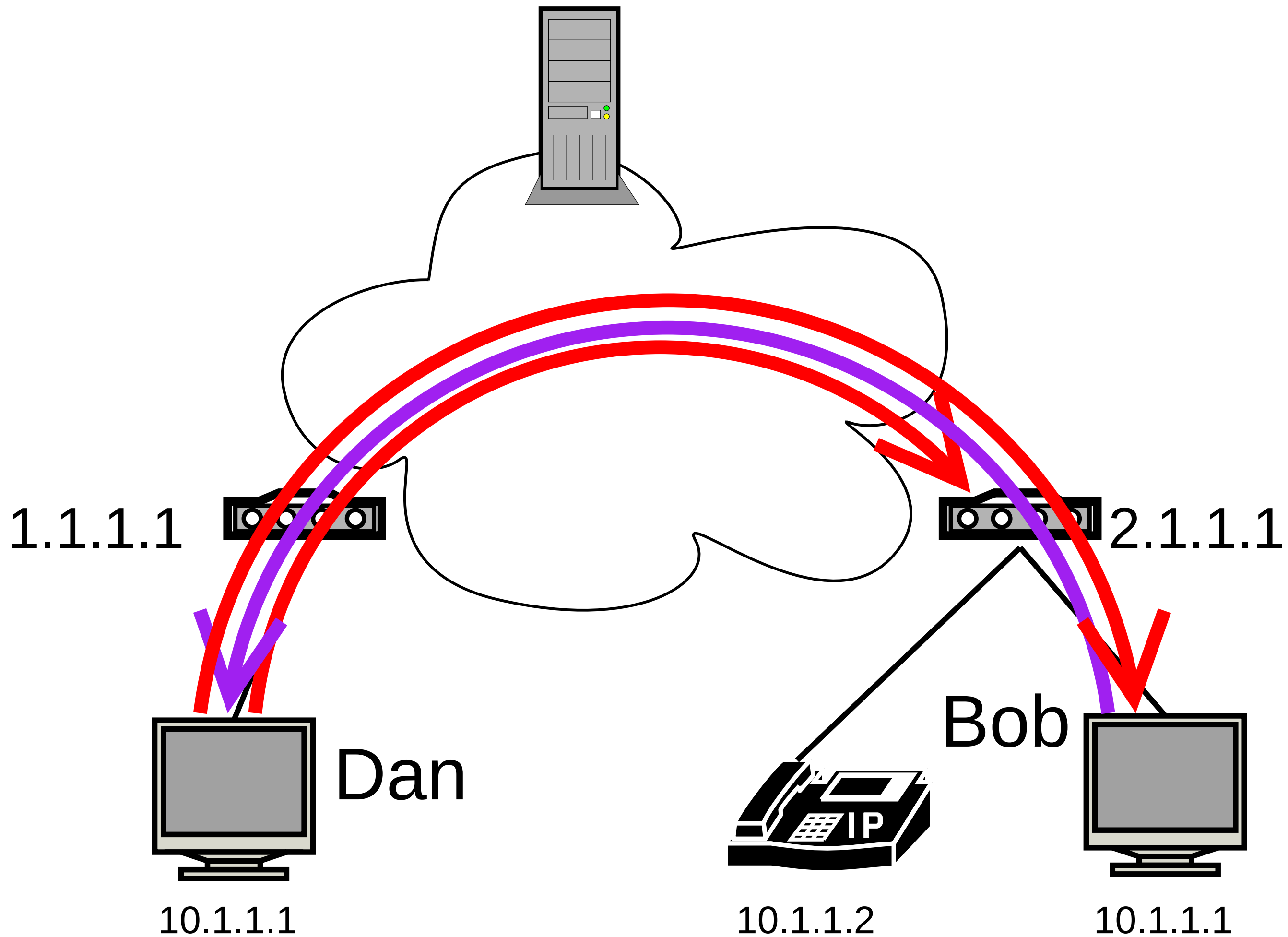
Basic solution for UDP

P2P connectivity through NATs



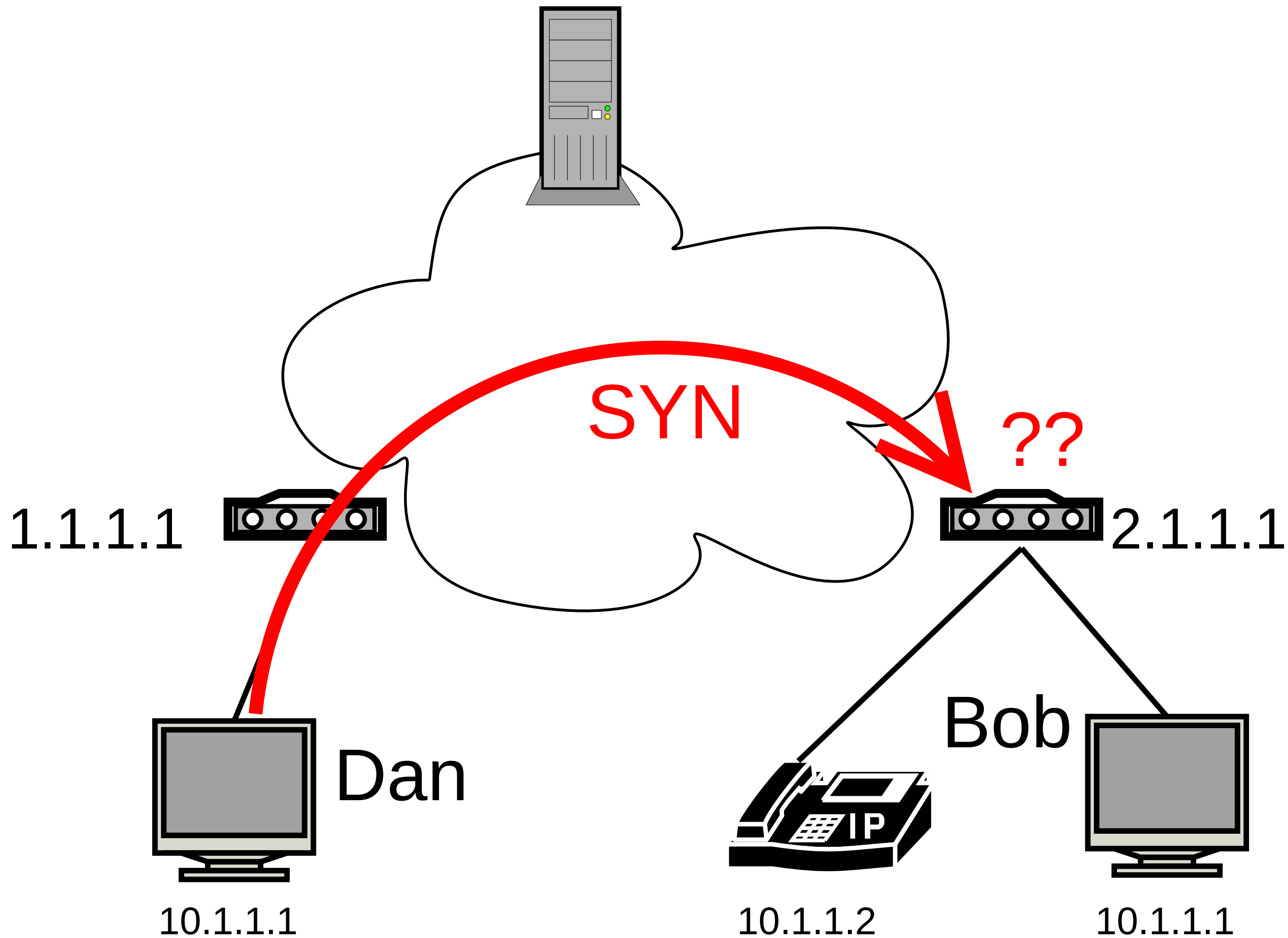
Basic solution for UDP

P2P connectivity through NATs



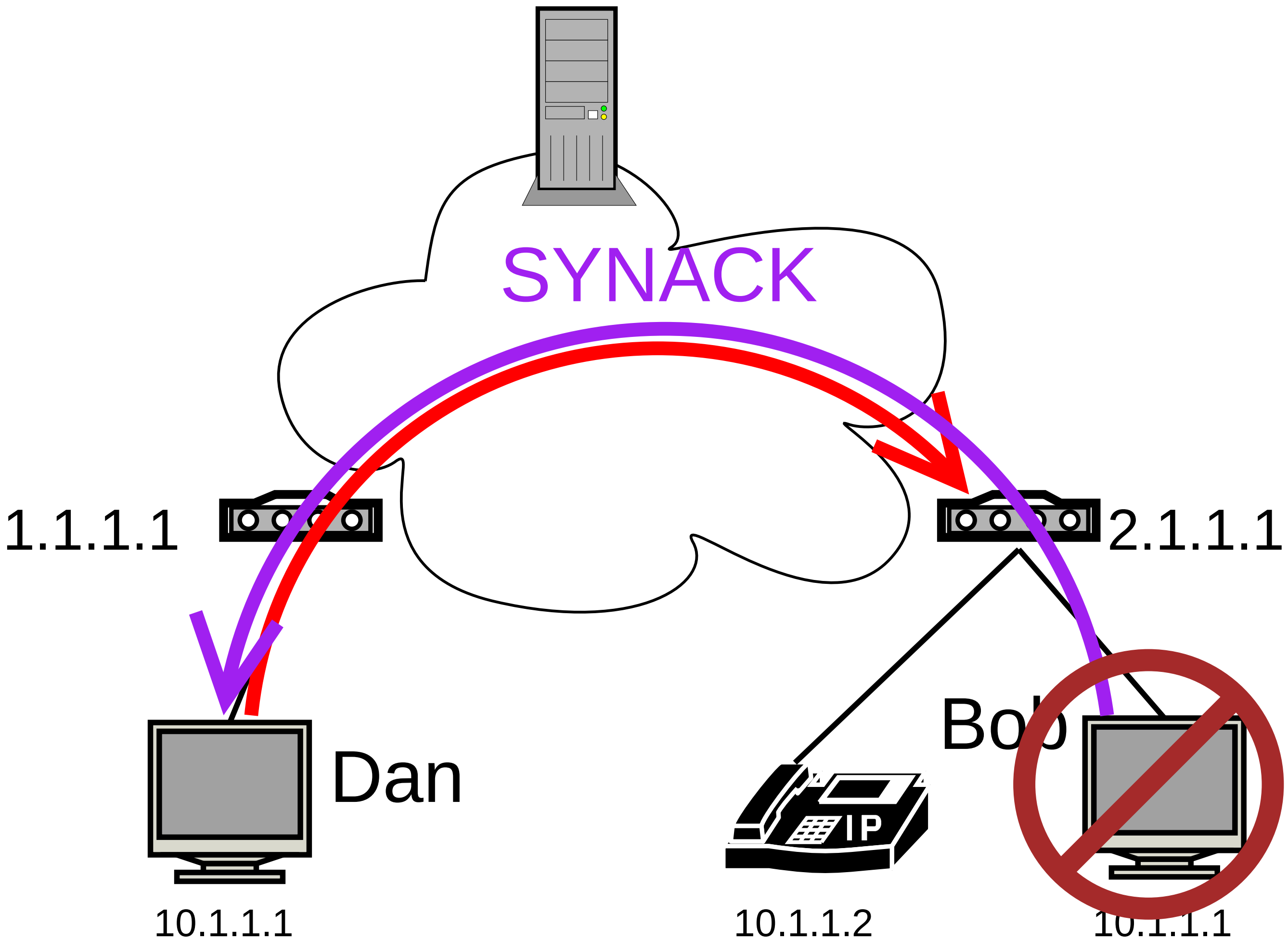
Basic solution for UDP

P2P connectivity through NATs



TCP establishment more complex

P2P connectivity through NATs



TCP establishment more complex

Context for this work

'92	NAT Invented
'93-'96	NAT traversal presumed impossible
'97-'01	UDP traversal solved and standardized [Kegel]
'93-'03	TCP traversal presumed impossible
'04	TCP traversal 'solved' (2 approaches) [Guha]
'05	2 more approaches [Ford, Biggadike]
'05	Approaches evaluated [Guha]
'06	TCP traversal standardized

Context for this work

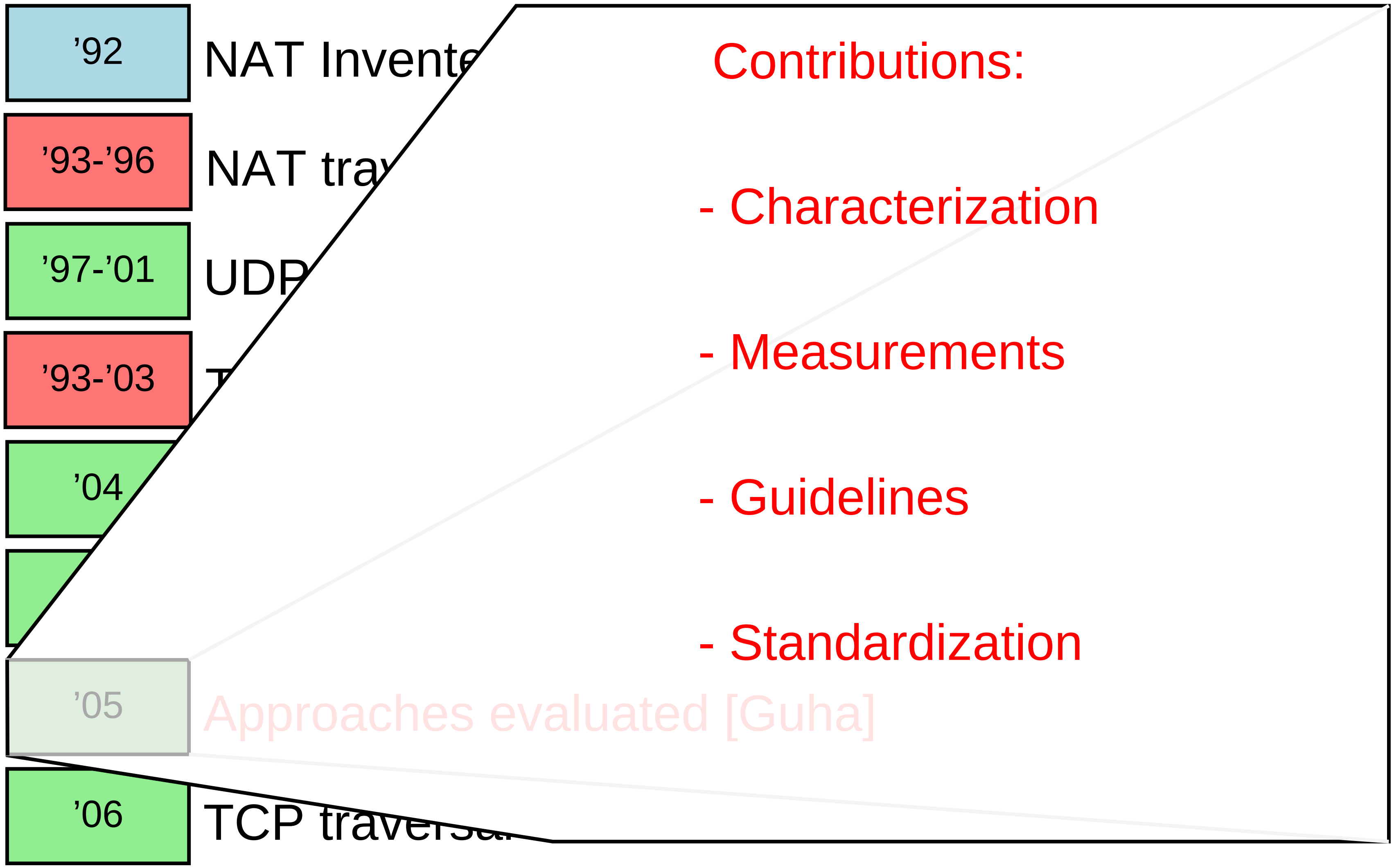
’92	NAT Invent
’93-’96	NAT t
’97-’01	
’9	
’04	TCP traversal ’solved’ (2 approaches) [Guha]
’05	2 more approaches [Ford, Biggadike]
’05	
’06	TCP traversal

4 approaches

Many trade-offs

- NAT sensitivity
- Ease of Implementation
- Ease of Deployment

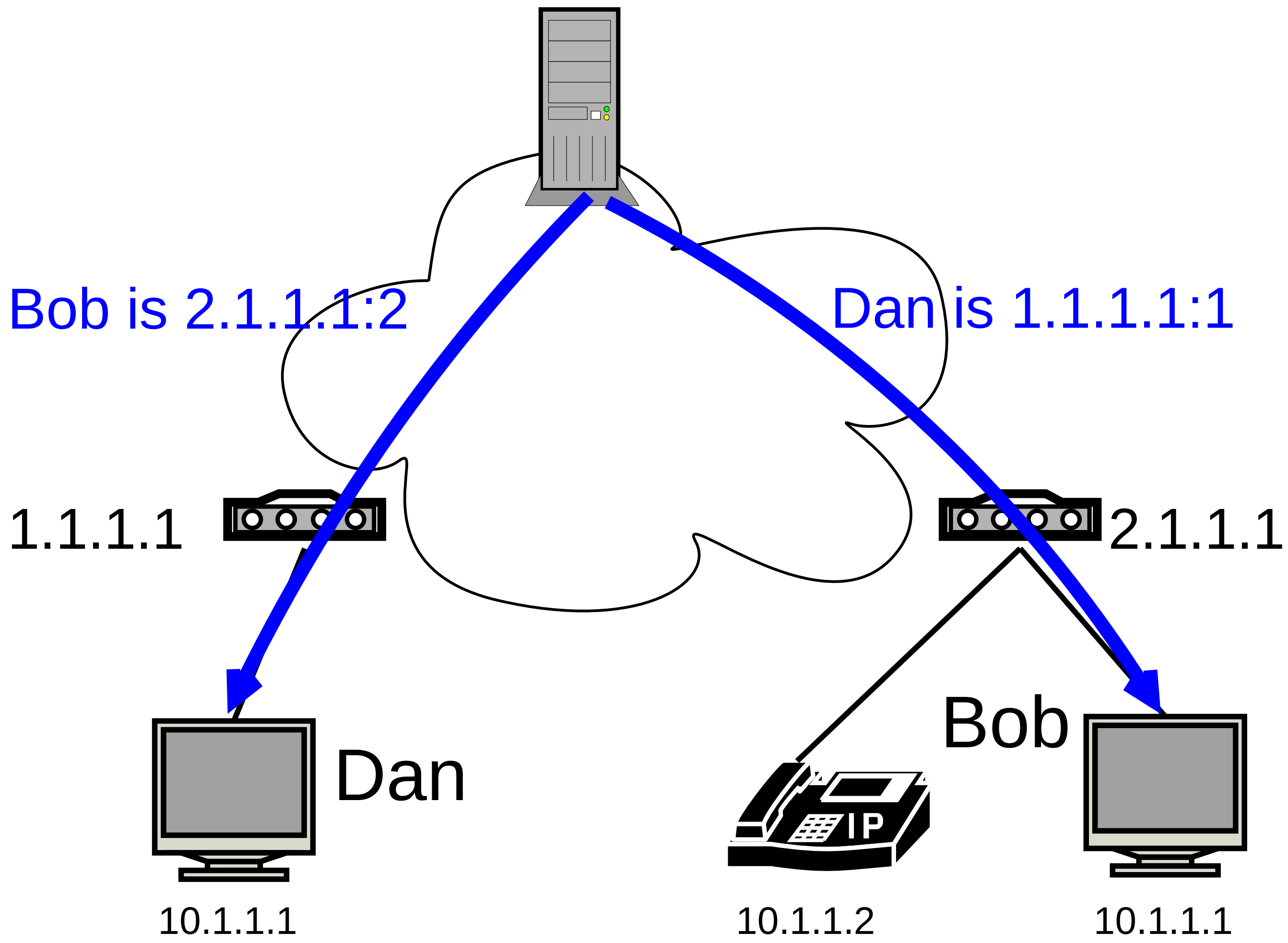
Context for this work



“Take away” Results

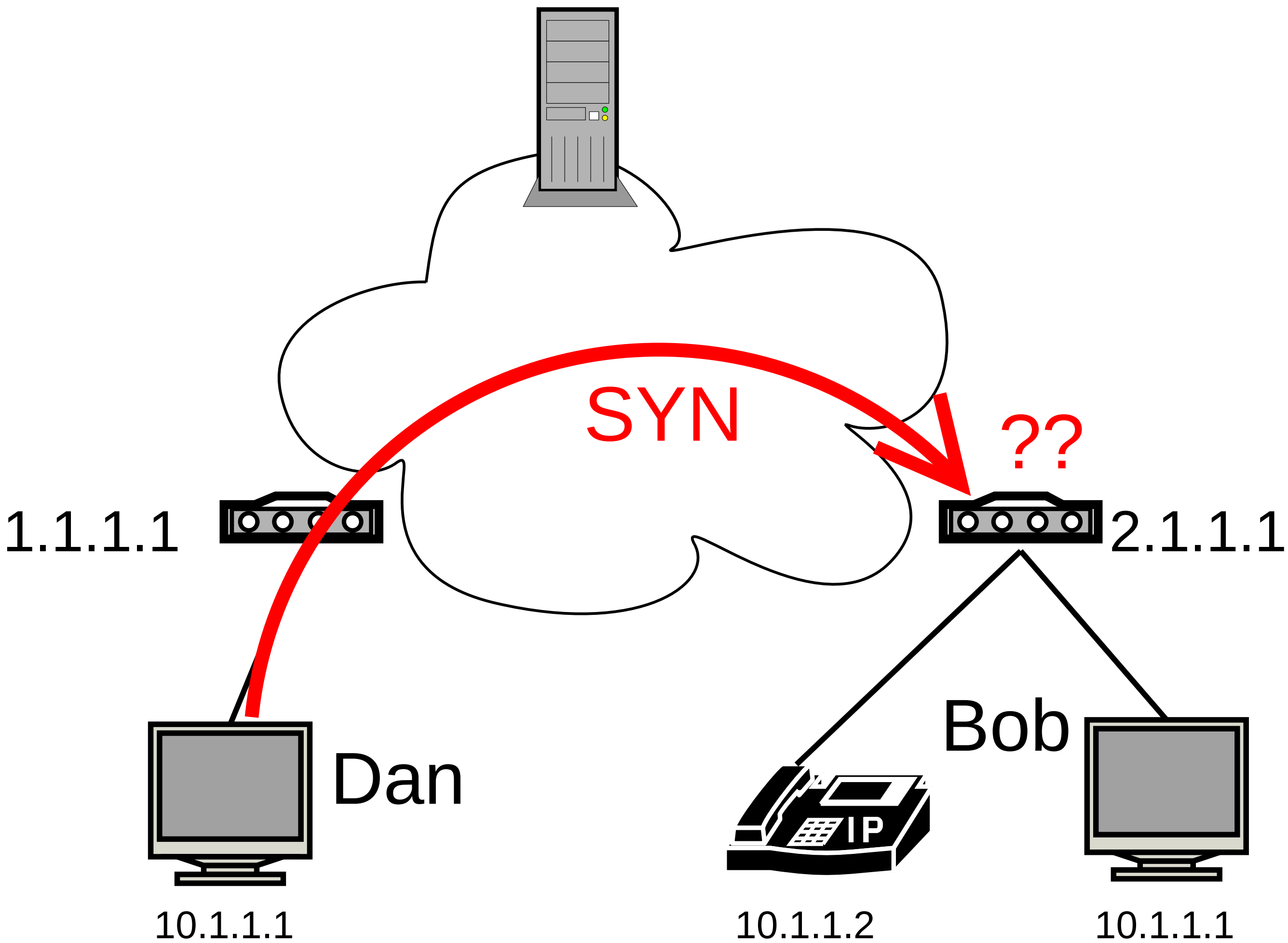
- TCP can be established between NAT’ed peers
- Works an estimated 85%–90% of the time today
- 100% for certain popular, well-behaved NATs
    - All NATs could standardize to this

P2P TCP Establishment



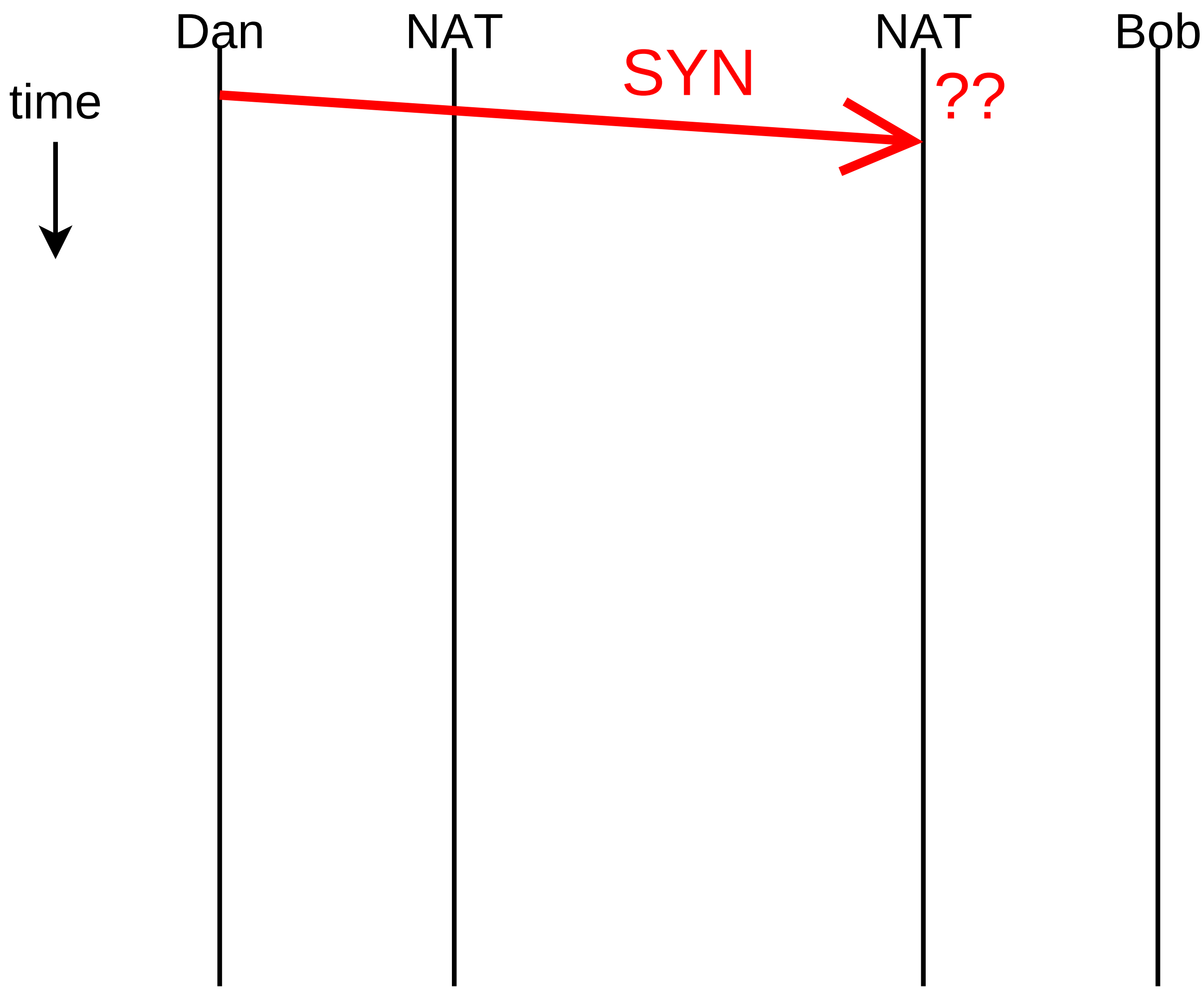
Use Rendezvous Service

P2P TCP Establishment



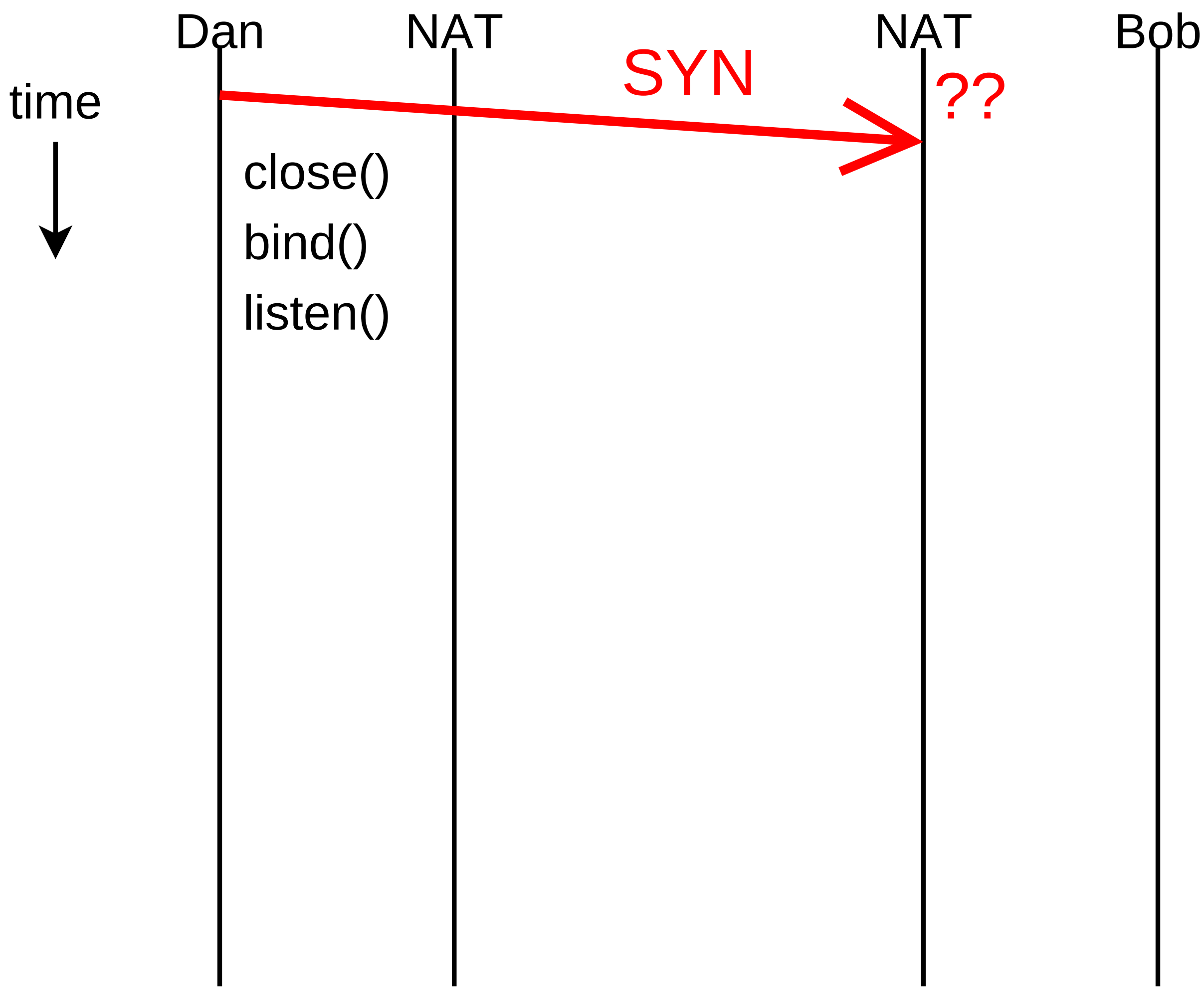
Use Rendezvous Service

P2P TCP Establishment



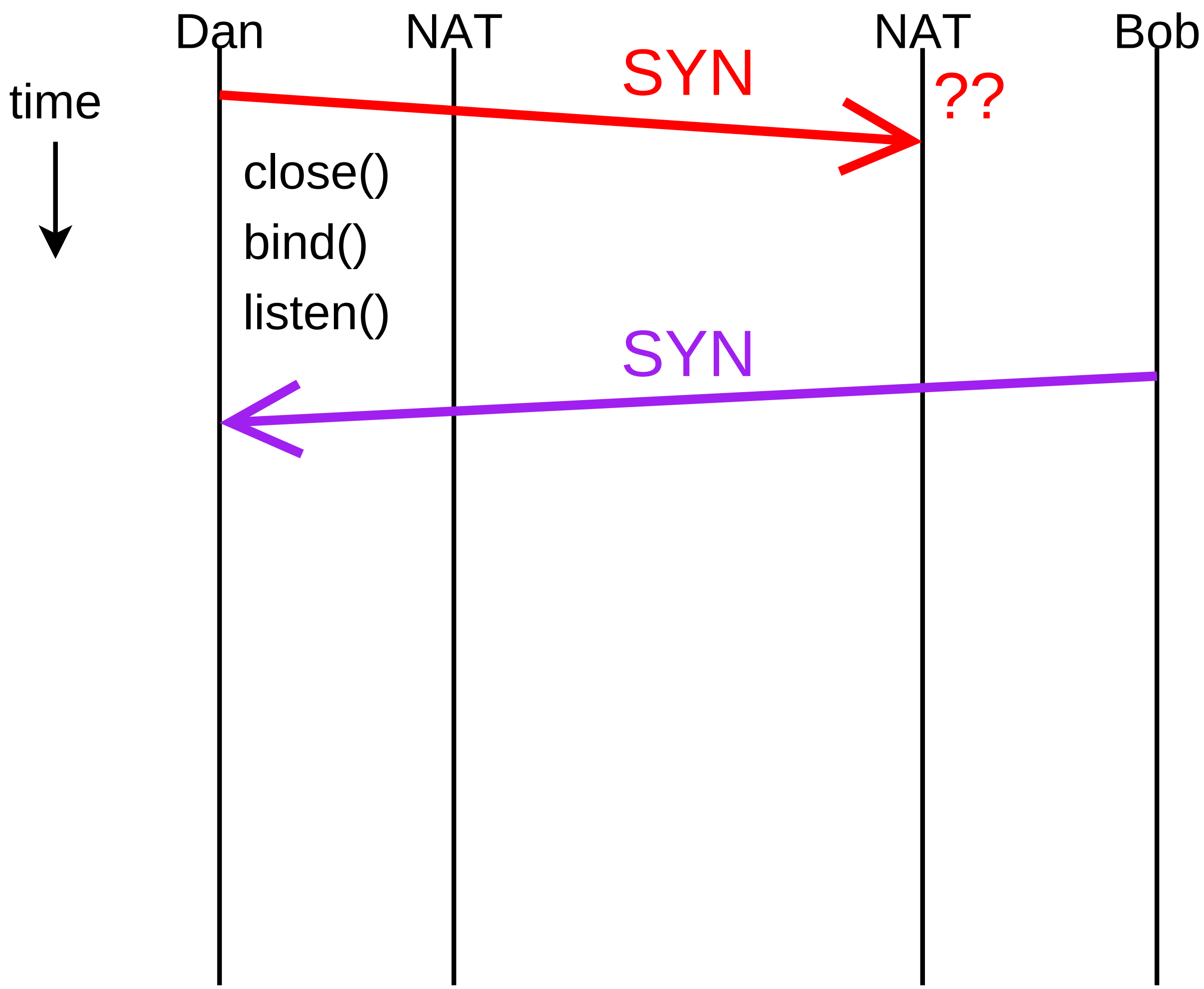
Punch hole using connect/close/bind/listen

P2P TCP Establishment



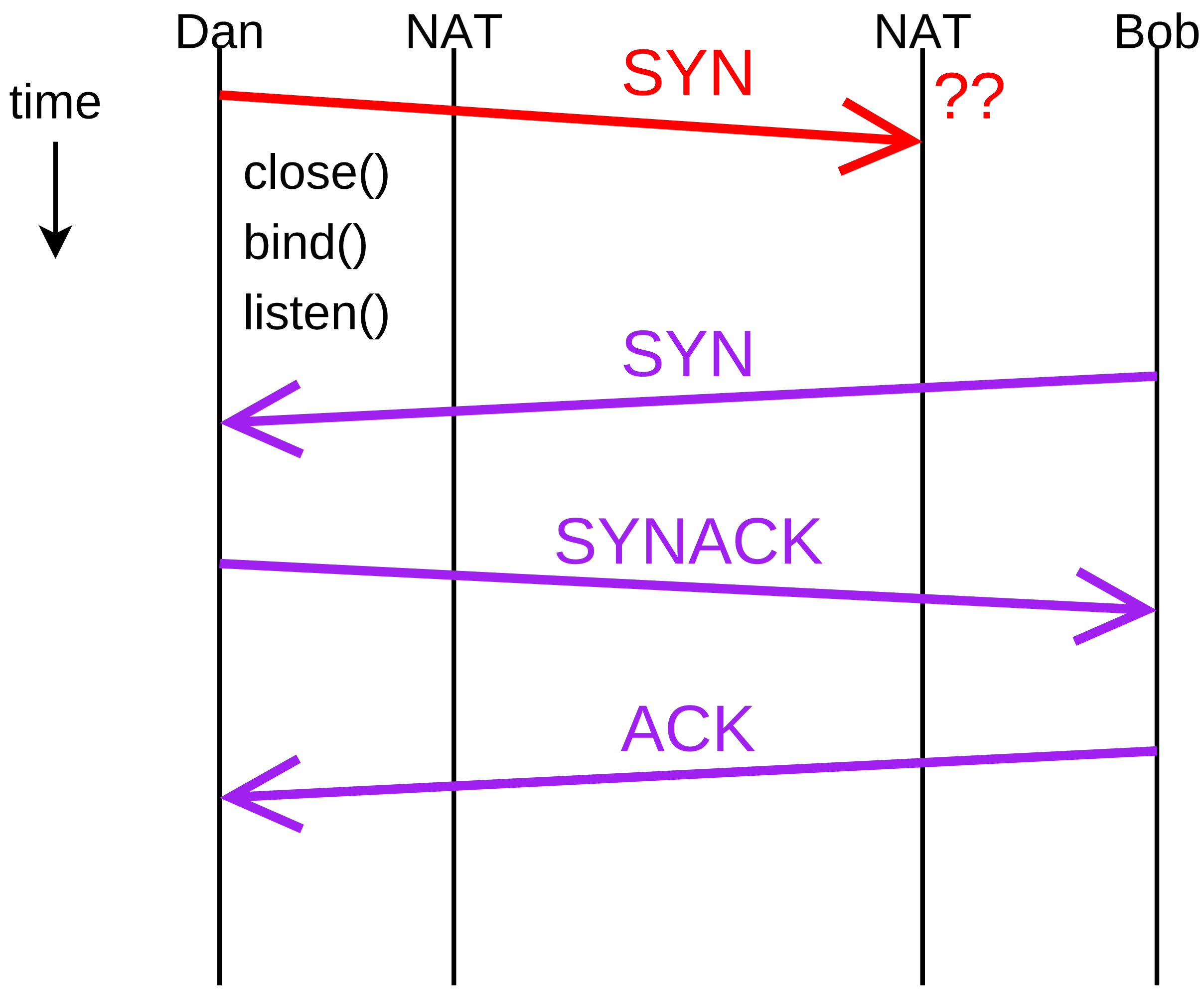
Punch hole using connect/close/bind/listen

P2P TCP Establishment



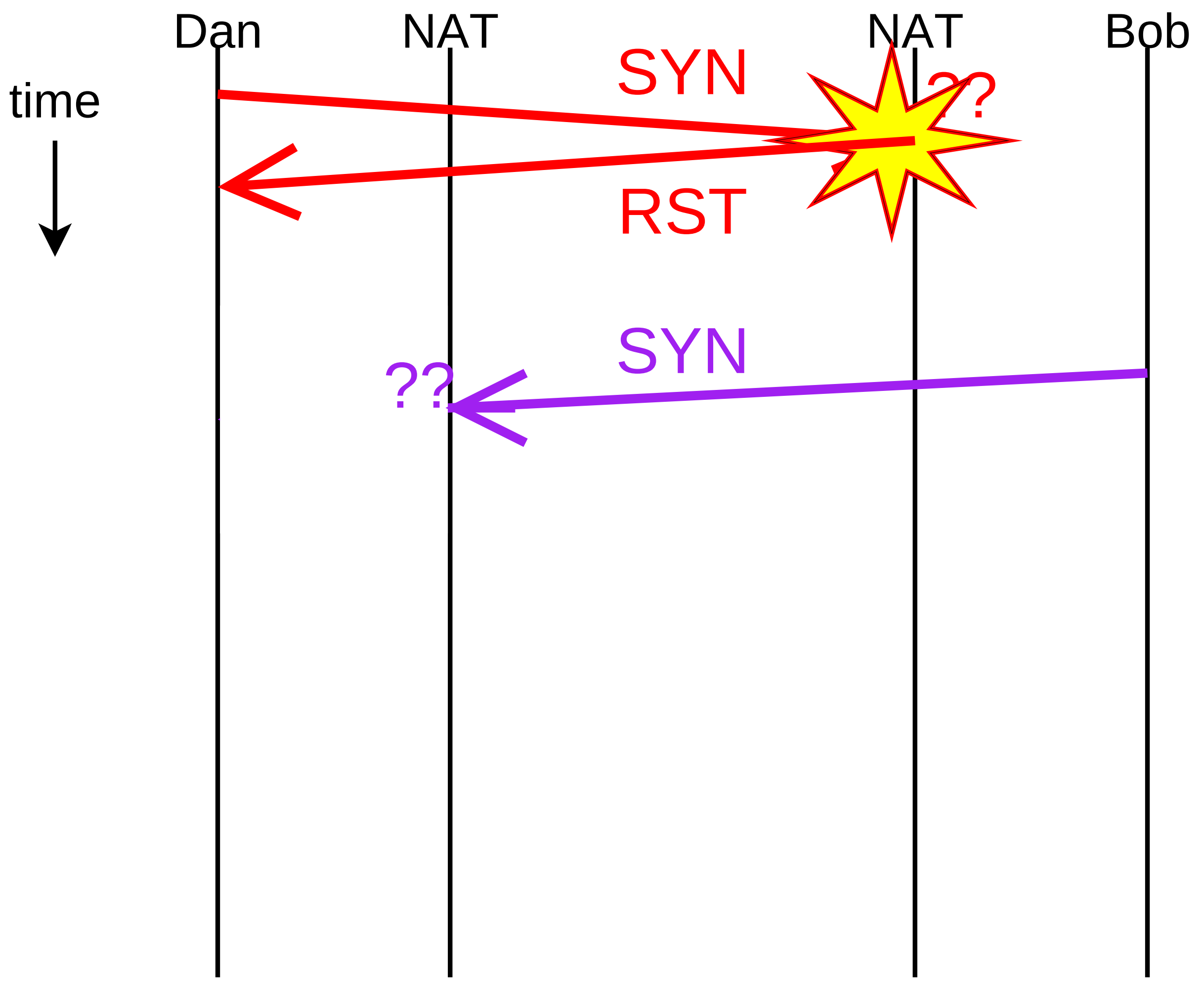
Accept incoming connection

P2P TCP Establishment



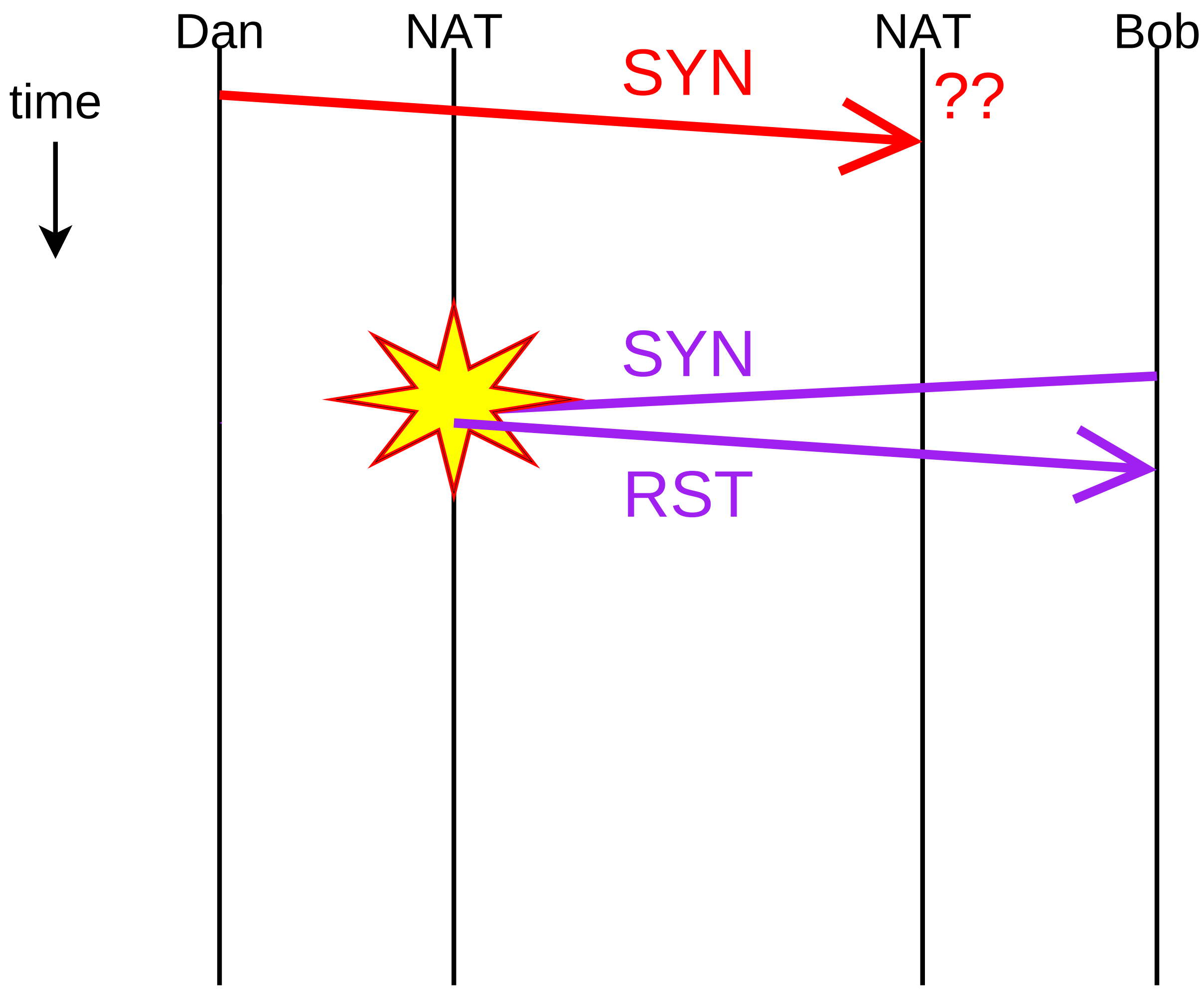
Accept incoming connection

P2P TCP Establishment



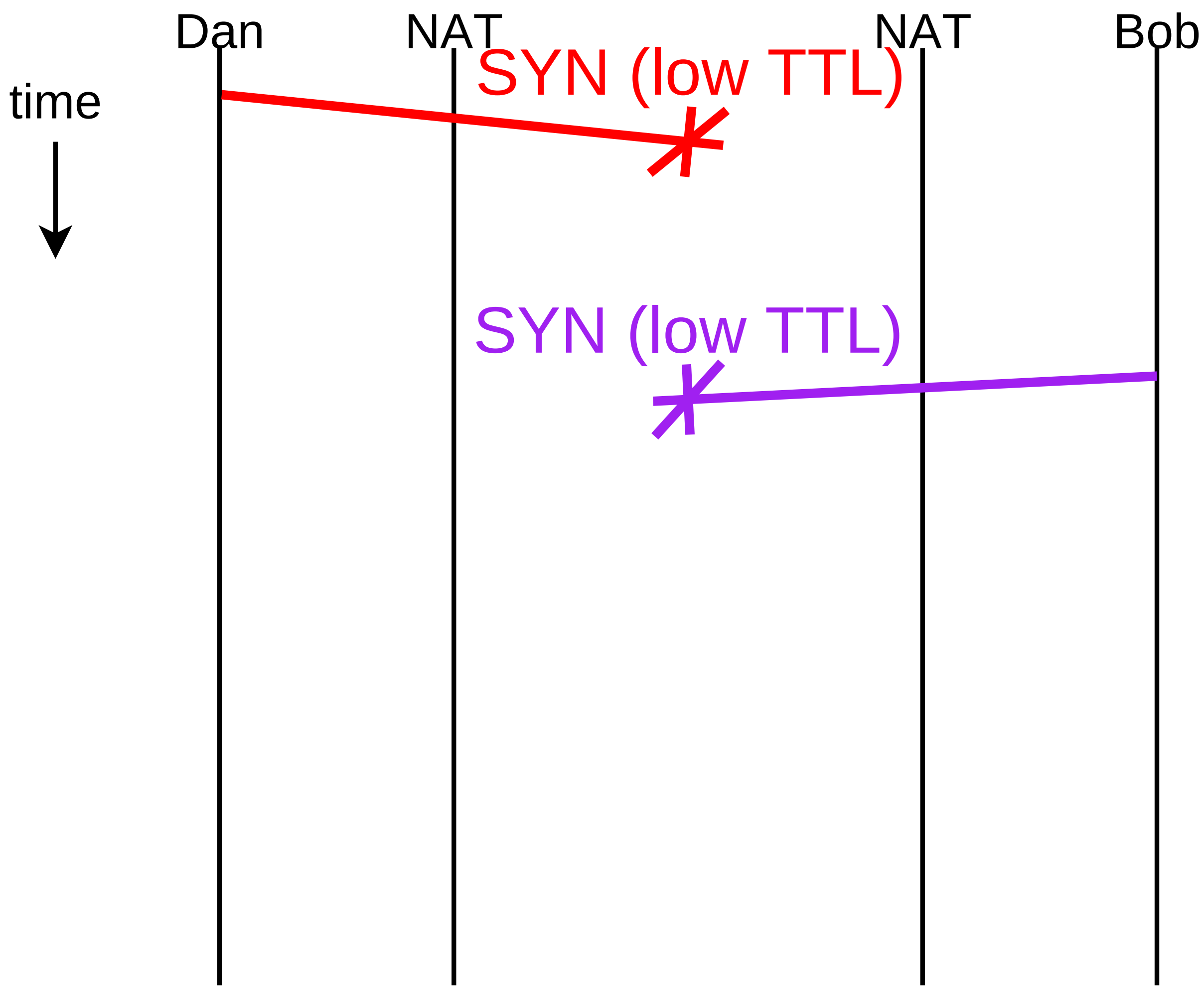
What if: NAT returns RST, closes hole

P2P TCP Establishment



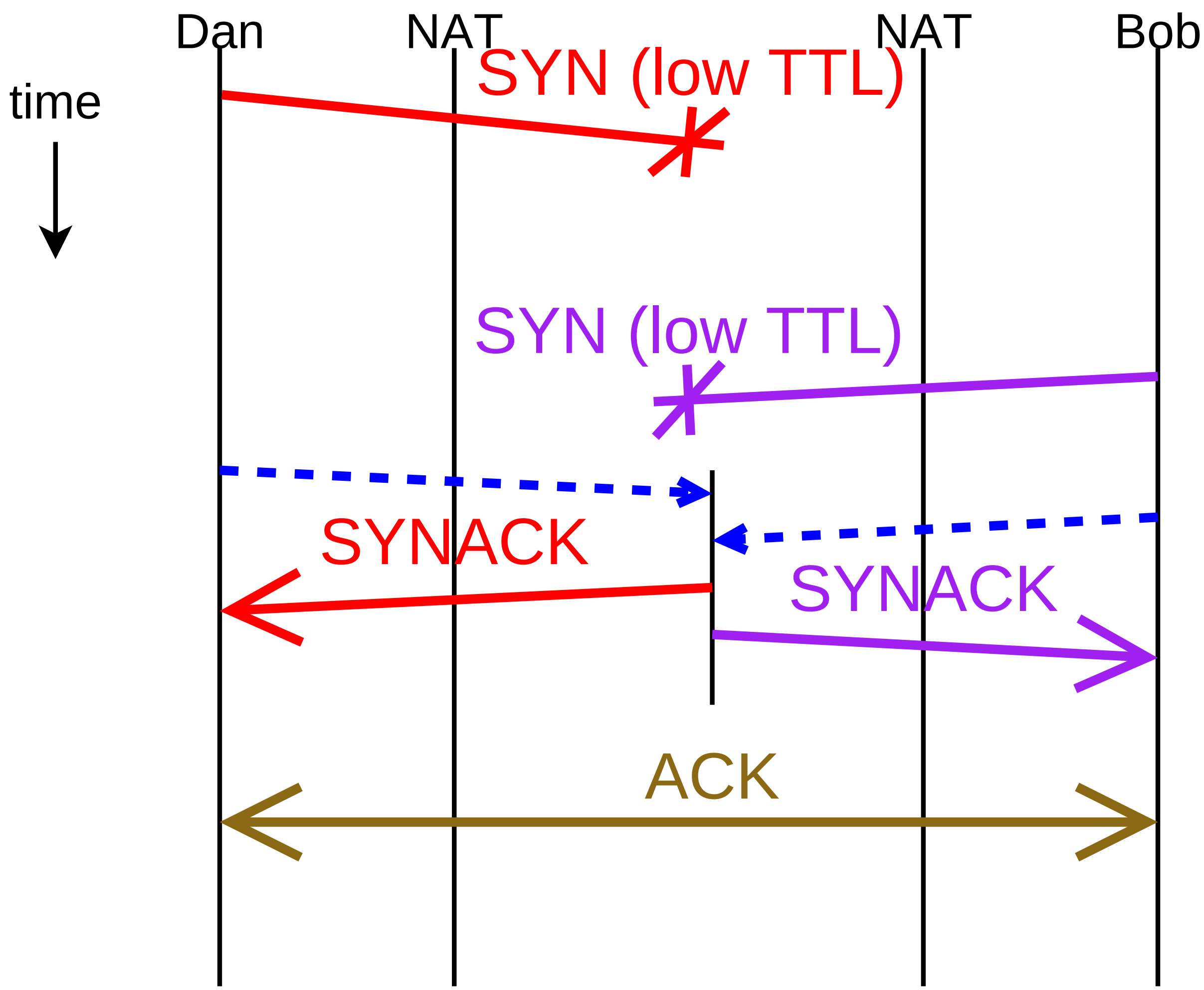
What if: NAT rejects SYN through hole

P2P TCP Establishment



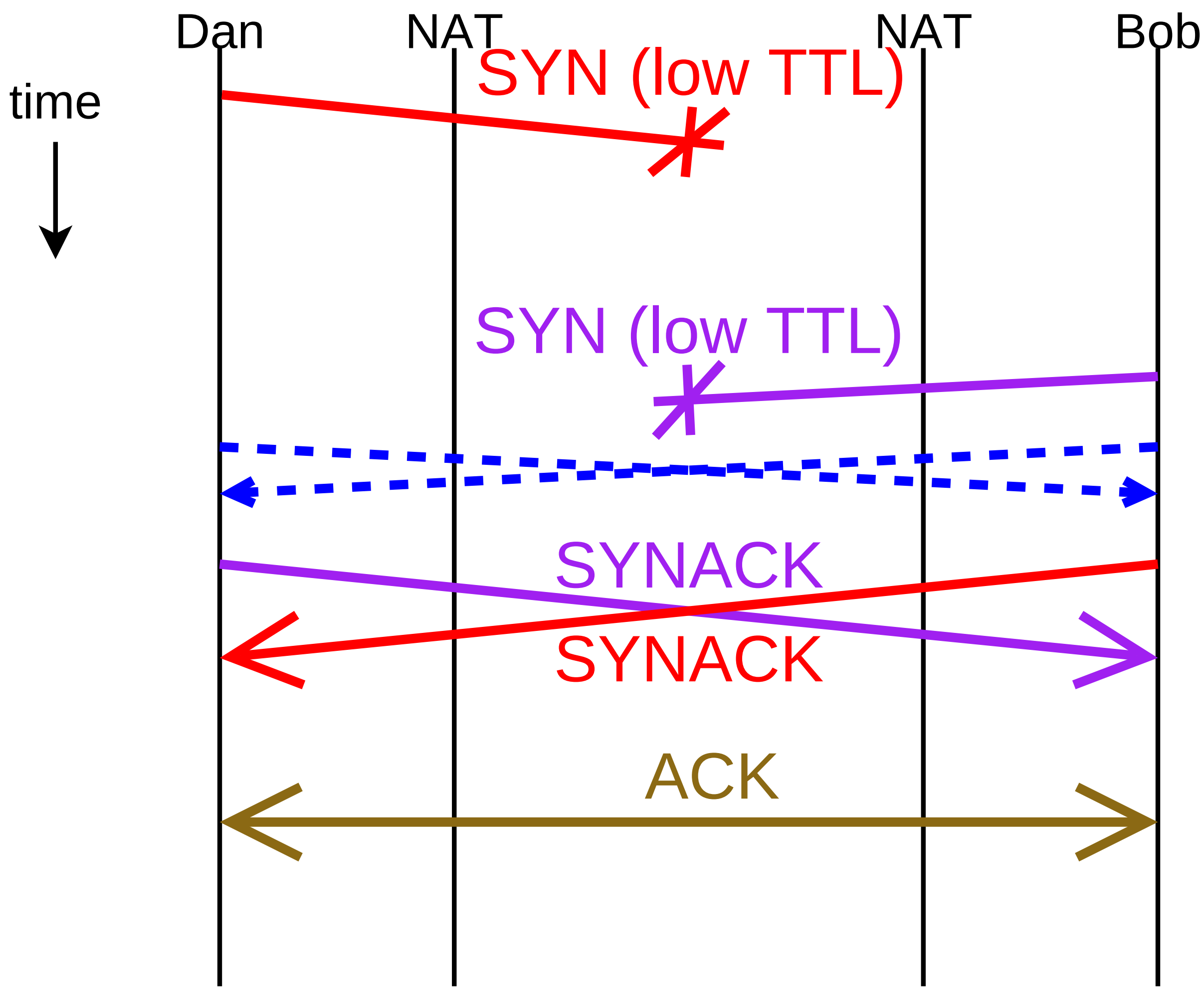
Variation: low-TTL SYN

P2P TCP Establishment



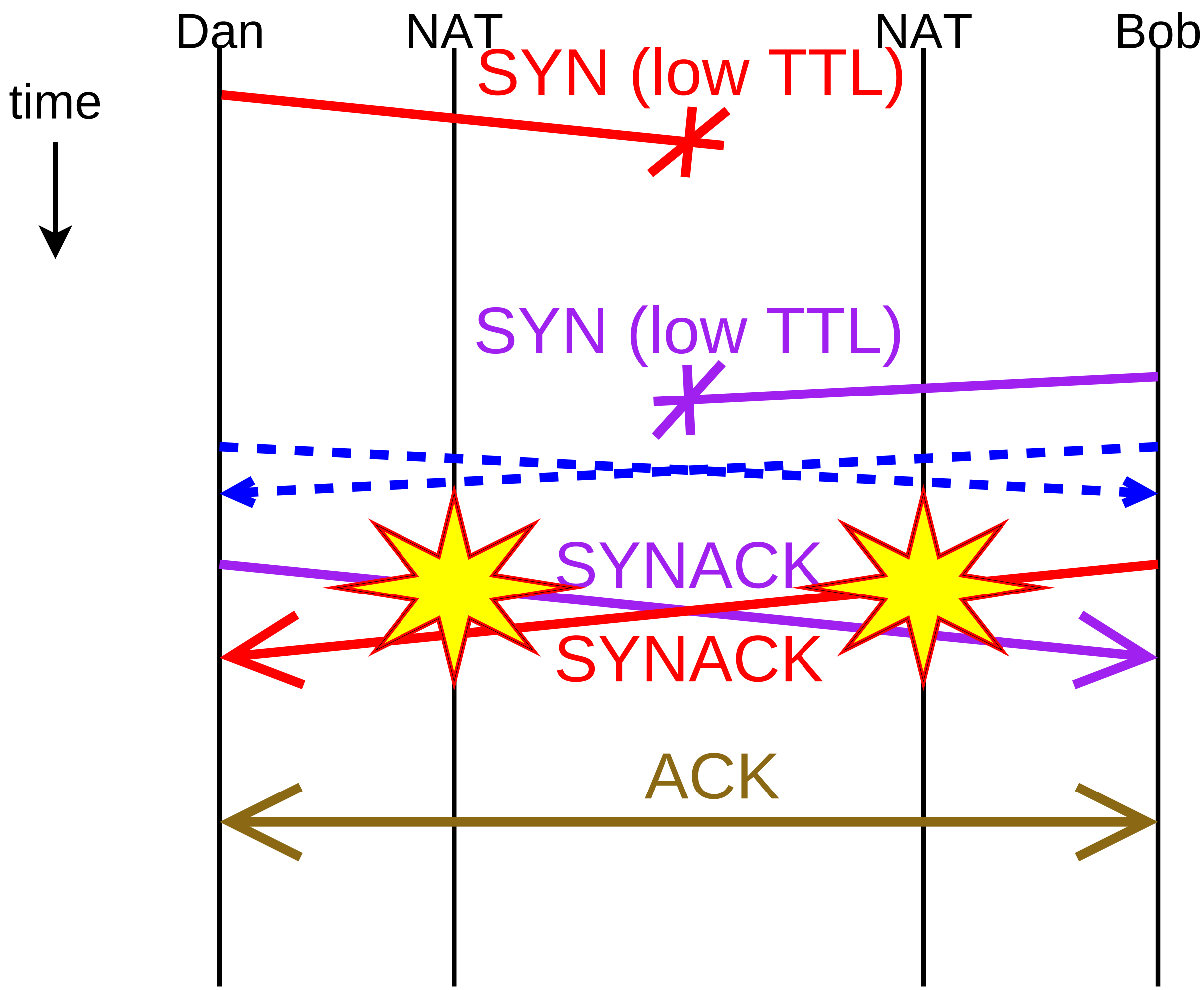
Variation: low-TTL SYN, spoof SYNACK

P2P TCP Establishment



Variation: low-TTL SYN, RAW SYNACK

P2P TCP Establishment



What if: NAT blocks outgoing SYNACK

Recap

- 4 approaches
  - 16 variants (mix and match)
- Many trade-offs
  - Some sensitive to NATs behavior
  - Some hard to implement
  - Some hard to deploy
- Measurement study to determine how well each works in practice

Methodology

- Implemented all approaches
  - Lessons learned in the paper
- Cause of failure for 16 brands of NATs
  - Linksys, DLink, Netgear, Belkin, . . .
- 32 axis of classification
- Classified ($\sim$100) NATs in the wild
- Extrapolated for world-wide behavior
  - Brand share market analysis

NAT Axes of Classification

NAT Binding:		
Type	Delta	Hairpin
Overloading	Max Flows	Predictable
Preservation:		
Port Number	Low	High
Dynamic	Parity	Sequential
Packet Mangling:		
TCP Data	ICMP Data	TCP Sequence
IP TTL		
Filters:		
$\overleftarrow{SYN}$	$\overleftarrow{SYN}$ (known IP)	Estd. $\overleftarrow{SYN}$
$\overrightarrow{SYN}$ $\overleftarrow{SYN}$	$\overrightarrow{SYN}$ $\overleftarrow{RST}$ $\overleftarrow{SYN}$	$\overrightarrow{SYN}$ $\overleftarrow{ICMP11}$ $\overleftarrow{SYN}$
$\overrightarrow{SYN}$ $\overleftarrow{ICMP2}$ $\overleftarrow{SYN}$	$\overrightarrow{SYN}$ $\overleftarrow{SYNACK}$	$\overrightarrow{SYN}$ $\overleftarrow{RST}$ $\overleftarrow{SYNACK}$
$\overrightarrow{SYN}$ $\overleftarrow{ICMP11}$ $\overleftarrow{SYNACK}$	$\overrightarrow{SYN}$ $\overleftarrow{ICMP2}$ $\overleftarrow{SYNACK}$	$\overrightarrow{SYN}$ $\overrightarrow{SYNACK}$
Timers:		
SYN-SENT	Established	Timed-Wait
RST		

NAT Axes of Classification

NAT Binding:

Type	Delta	Hairpin
Overloading	Max Flows	Predictable

Preservation:

Port Number	Low	High
Dynamic	Parity	Sequential

Packet Mangling:

TCP Data	ICMP Data	TCP Sequence
IP TTL		

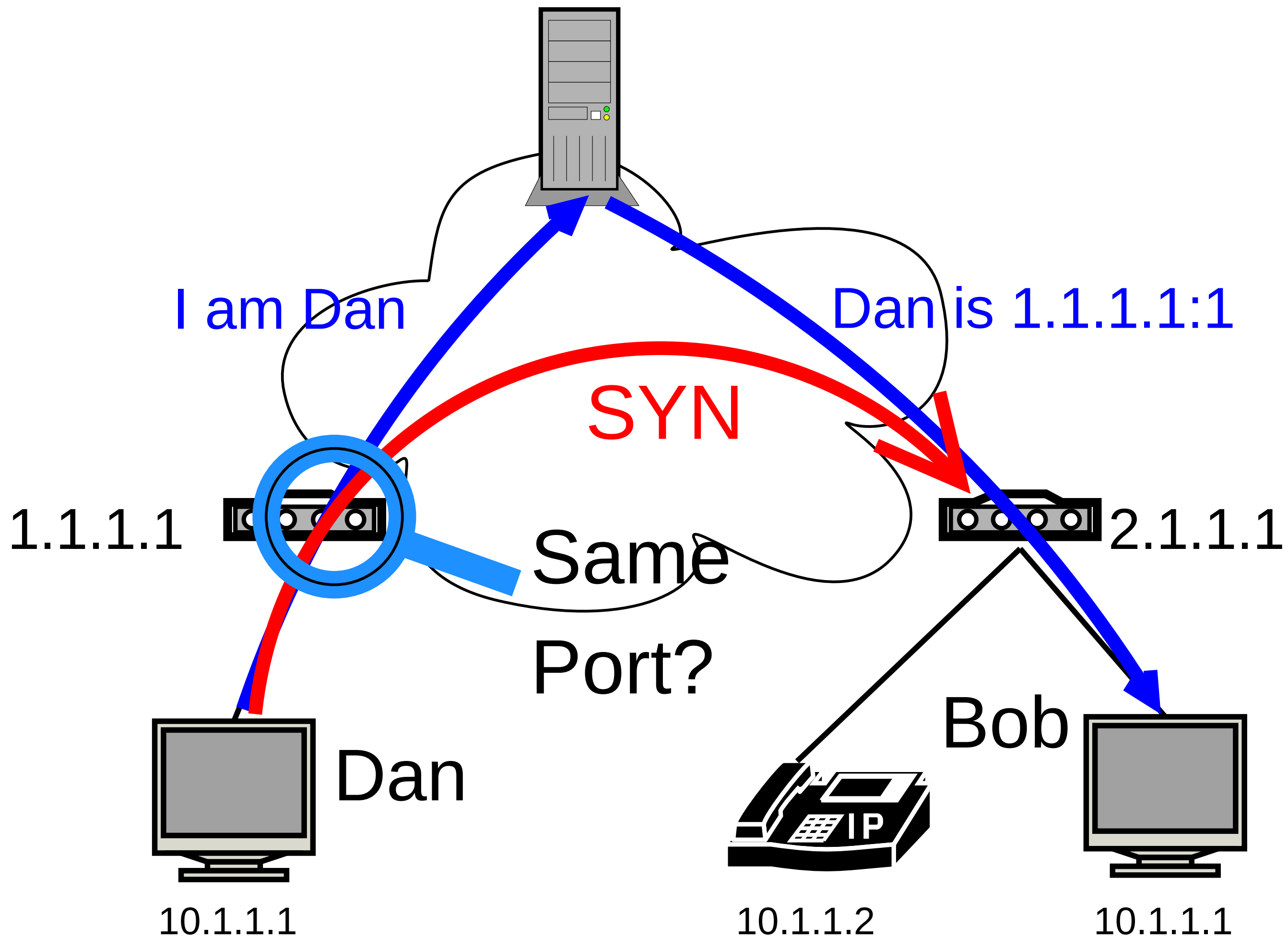
Filters:

$\overleftarrow{SYN}$	$\overleftarrow{SYN}$ (known IP)	Estd. $\overleftarrow{SYN}$
$\overrightarrow{SYN}$ $\overleftarrow{SYN}$	$\overrightarrow{SYN}$ $\overleftarrow{RST}$ $\overleftarrow{SYN}$	$\overrightarrow{SYN}$ $\overleftarrow{ICMP11}$ $\overleftarrow{SYN}$
$\overrightarrow{SYN}$ $\overleftarrow{ICMP2}$ $\overleftarrow{SYN}$	$\overrightarrow{SYN}$ $\overleftarrow{SYNACK}$	$\overrightarrow{SYN}$ $\overleftarrow{RST}$ $\overleftarrow{SYNACK}$
$\overrightarrow{SYN}$ $\overleftarrow{ICMP11}$ $\overleftarrow{SYNACK}$	$\overrightarrow{SYN}$ $\overleftarrow{ICMP2}$ $\overleftarrow{SYNACK}$	$\overrightarrow{SYN}$ $\overrightarrow{SYNACK}$

Timers:

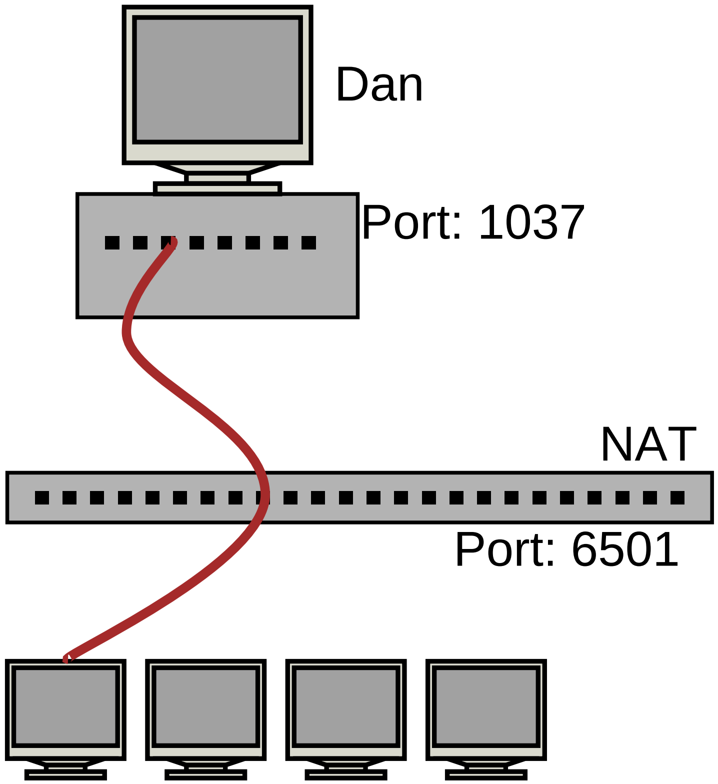
SYN-SENT	Established	Timed-Wait
RST		

Port Prediction



Problem: What port did SYN come from?

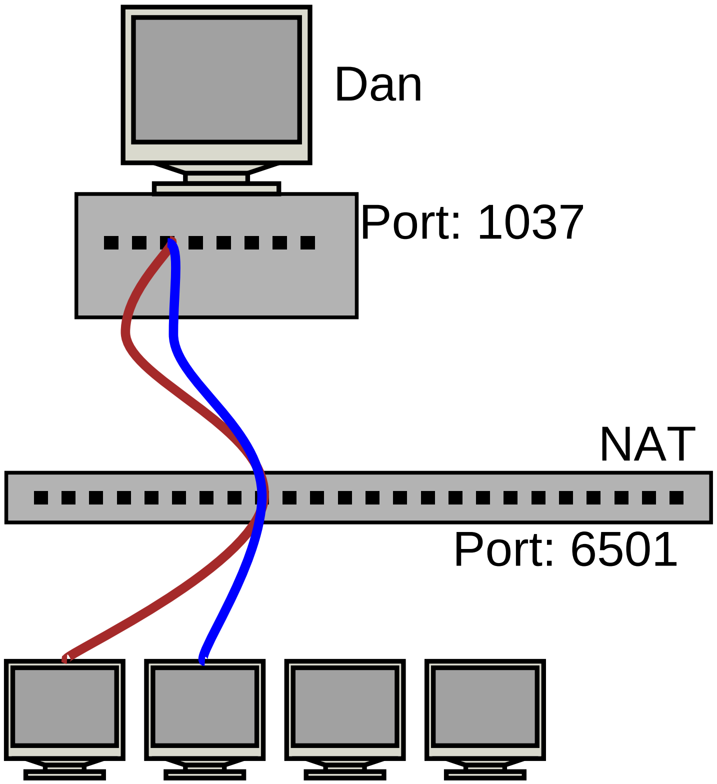
Port Prediction



Classification

NB:Independent

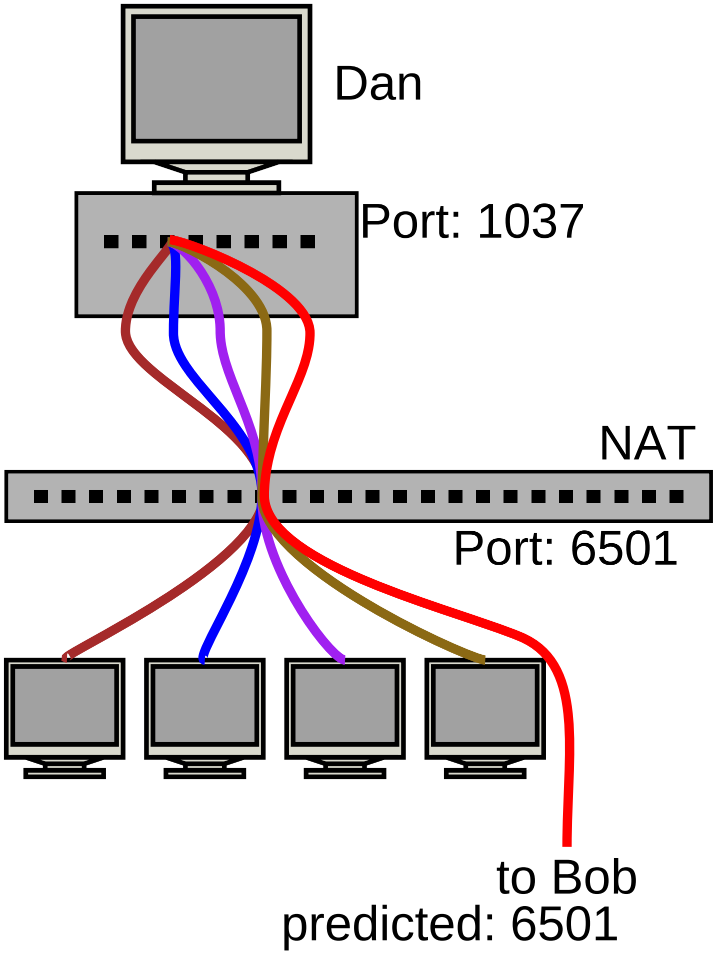
Port Prediction



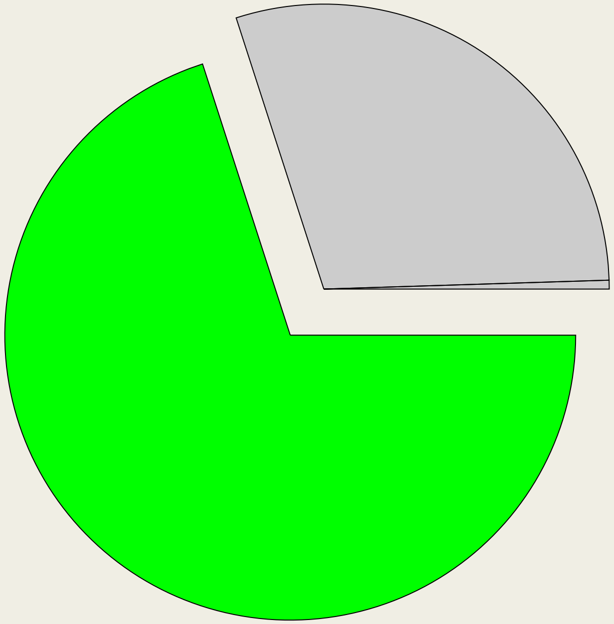
Classification

NB:Independent

Port Prediction

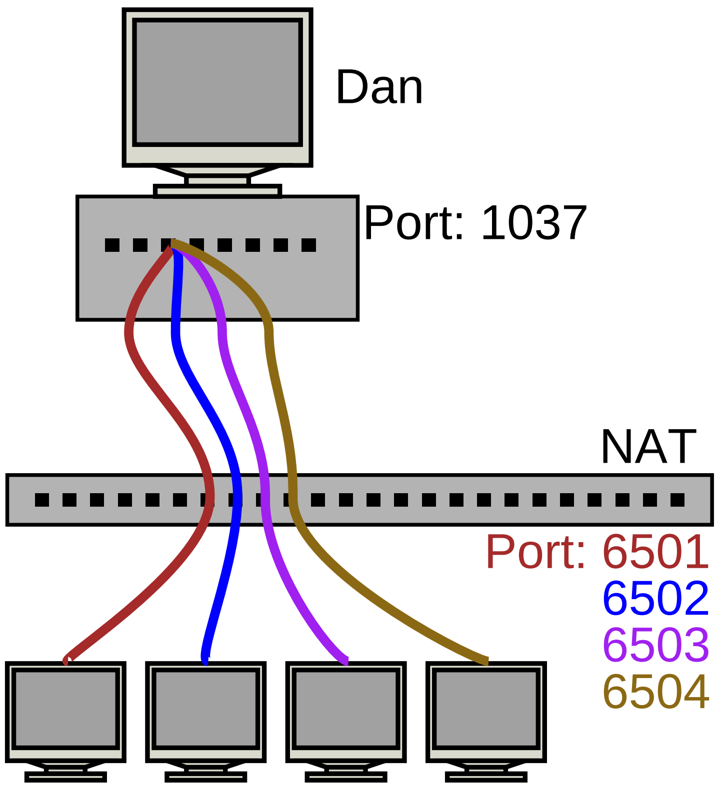


Classification

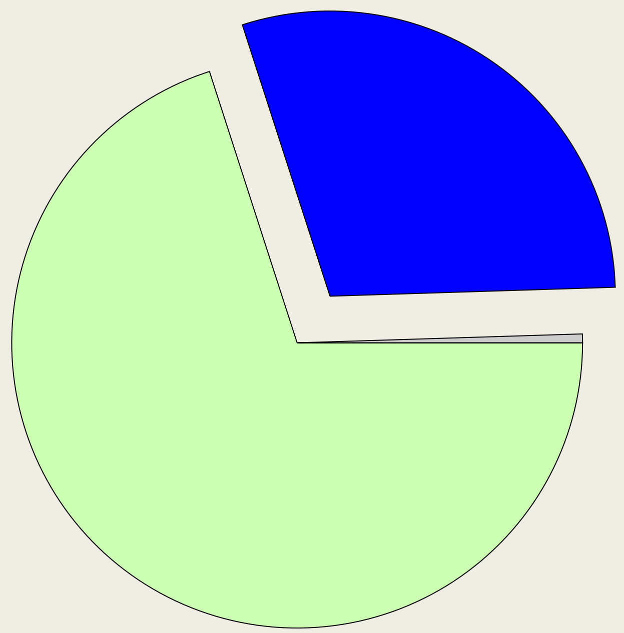


NB:Independent

Port Prediction

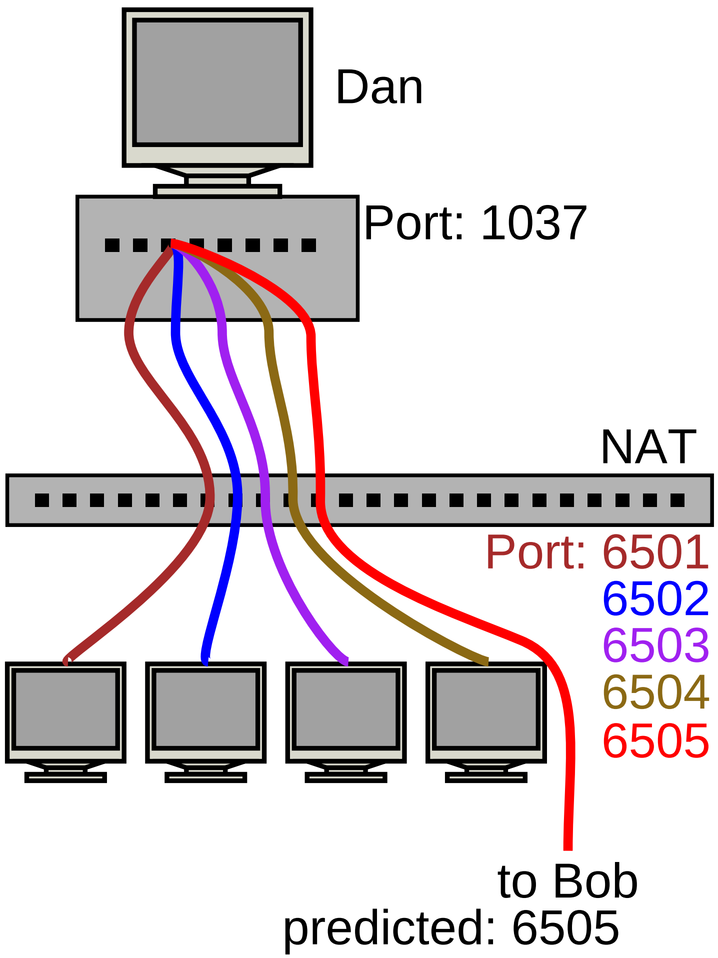


Classification

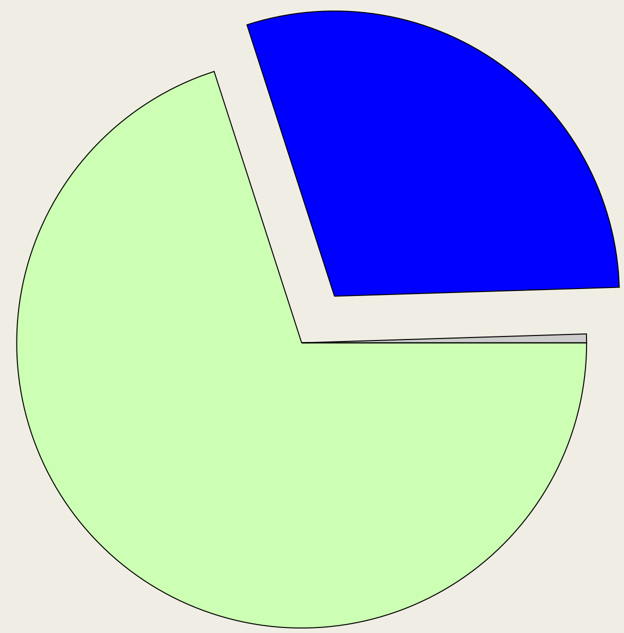


NB:Delta

Port Prediction

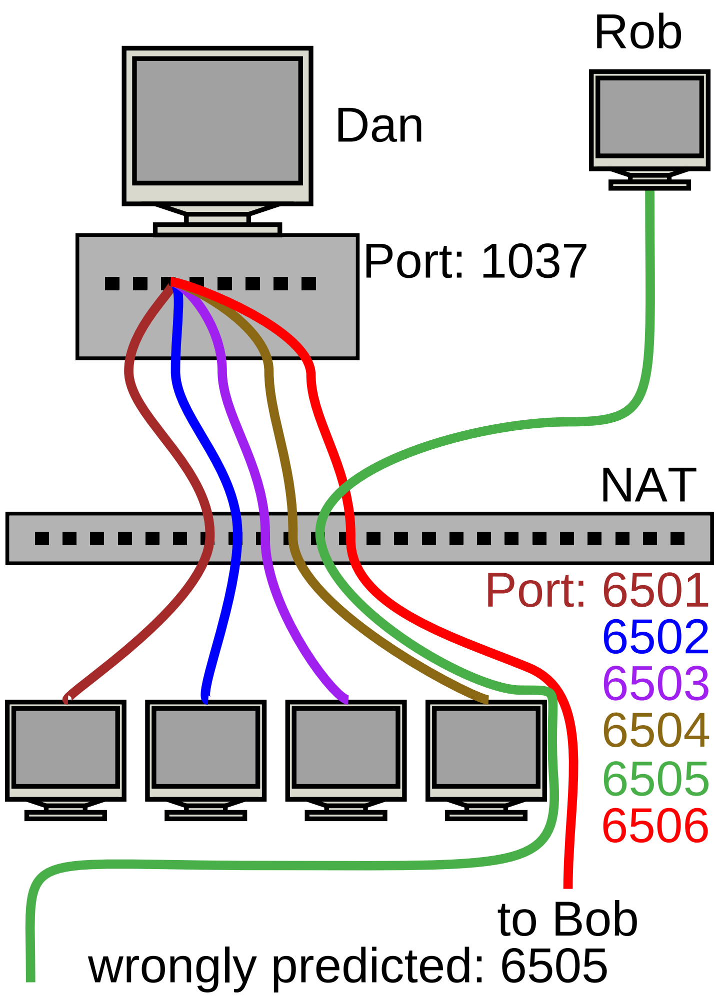


Classification

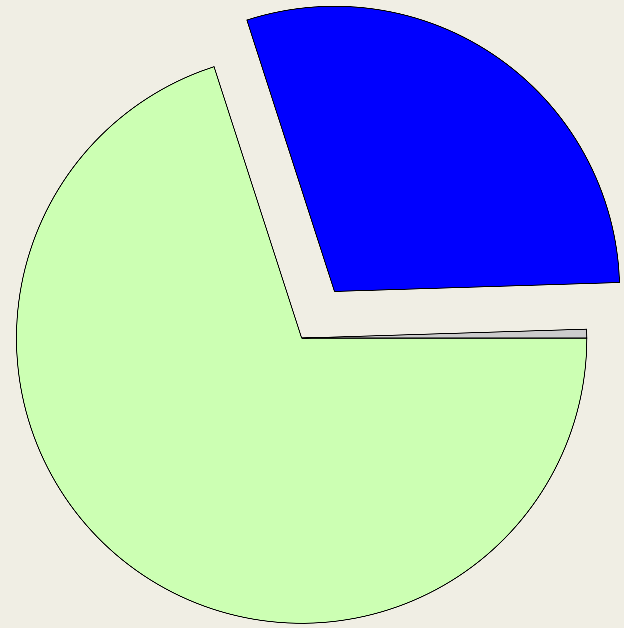


NB:Delta

Port Prediction

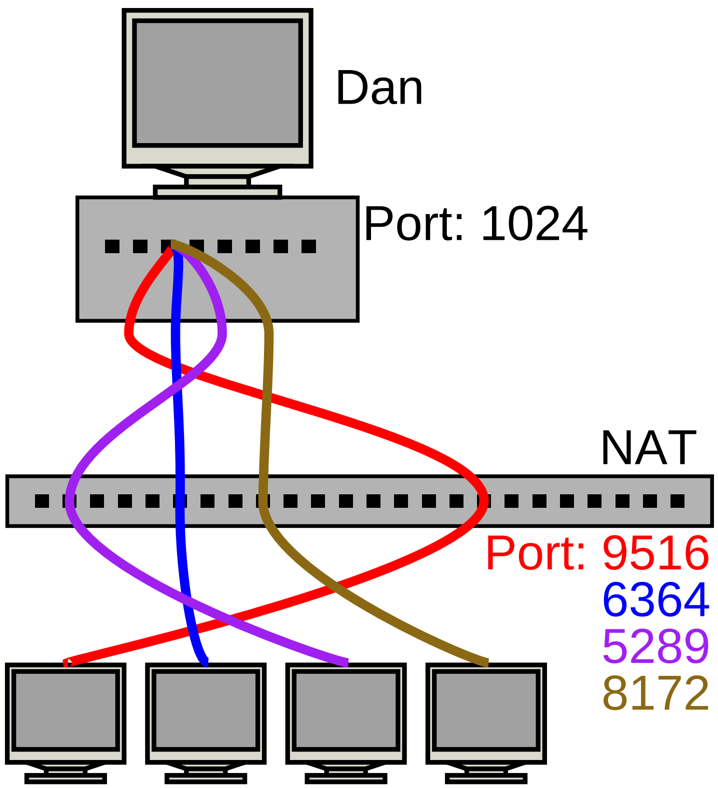


Classification

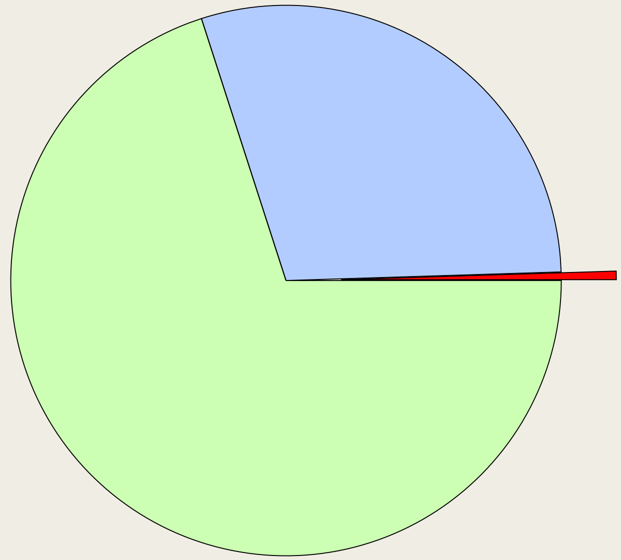


NB:Delta

Port Prediction

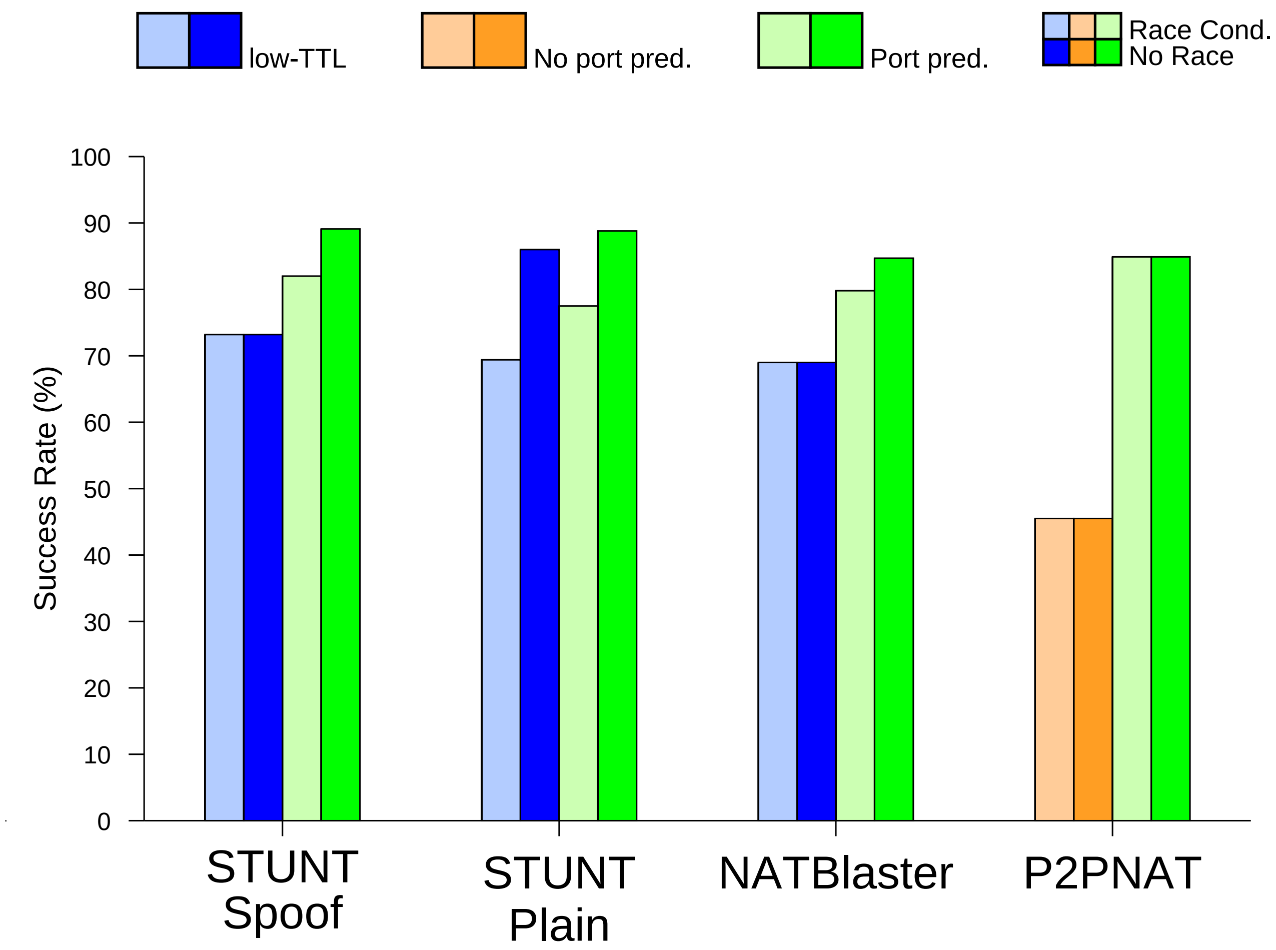


Classification



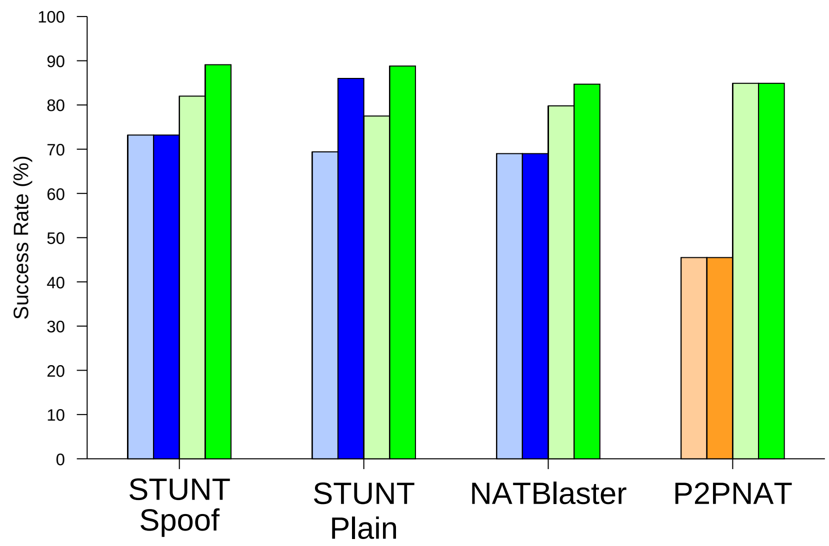
NB:Random

Projected Success



TCP traversal succeeds 85%-90% (estd.)

Projected Success



1. STUNT Spoof – Hard to deploy
2. STUNT Plain – Best Option
3. NATBlaster – Fails on WinXP SP2
4. P2PNAT – Fails on WinXP and earlier

Software

- NAT Traversal Library
  - JAVA implementation available
  - Encrypted tunnel application
- NAT Classification software
  - Windows, Linux versions available

Future Work

- Wide-scale testing
  - Implement in bittorrent, swarmcast, ...
- Standardize NAT TCP Behavior
  - IETF BEHAVE Working Group
  - I-D: draft-hoffman-behave

Related Issues

IPv6 . . .

- Transition will require v4–v6 NATs

Firewalls . . .

- Will persist even with IPv6

Universal Plug-and-Play (UPnP) . . .

- Off by default

Summary

- TCP NAT Traversal works!
  - 85%-90% today, 100% soon
- For P2P developers:
  - Application guidelines
  - TCP traversal library
- For NAT vendors:
  - Standards document
  - NAT checking software

`http://nutss.net/stunt`