The Need for Collaboration between ISPs and P2P

P2P systems from an ISP view

- **Structured**
  - DHTs (distributed hash tables), e.g., Chord, Pastry, Tapestry, CAN, Tulip, ...
  - Globally consistent protocol with “efficient” search
  - Ignores the underlay, arbitrary placement of “data”
  - Inefficient routing (log n is no good)

- **Unstructured**
  - Arbitrary neighbors, e.g., Gnutella, FastTrack, ...
  - Ignores the underlay, neighbor selection, download location selection
  - Inefficient routing
  - Does its own “traffic engineering”
P2P traffic

- Some source claim >50% of Internet traffic
  - Examples: Bittorrent, eDonkey, Skype, GoogleTalk...

![Internet traffic distribution 2007 (Germany)](source:image)

![Flow diagram](source:image)

Application Detection
Problem application detection

- Usually only by port number!
- Yet applications use arbitrary ports
  Benign reasons and malicious reasons
- Example:
  Network Intrusion Detection Systems

Ports accounting > 1% of conns.

<table>
<thead>
<tr>
<th>Port</th>
<th>% Conns</th>
<th>% Success</th>
<th>% Payload</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web 80</td>
<td>70.82%</td>
<td>68.13%</td>
<td>72.59%</td>
</tr>
<tr>
<td>445</td>
<td>3.53%</td>
<td>0.01%</td>
<td>0.00%</td>
</tr>
<tr>
<td>Web 443</td>
<td>2.34%</td>
<td>2.08%</td>
<td>1.29%</td>
</tr>
<tr>
<td>SSH 22</td>
<td>2.12%</td>
<td>1.75%</td>
<td>1.71%</td>
</tr>
<tr>
<td>Mail 25</td>
<td>1.85%</td>
<td>1.05%</td>
<td>1.71%</td>
</tr>
<tr>
<td>1042</td>
<td>1.66%</td>
<td>0.00%</td>
<td>0.00%</td>
</tr>
<tr>
<td>1433</td>
<td>1.06%</td>
<td>0.00%</td>
<td>0.00%</td>
</tr>
<tr>
<td>135</td>
<td>1.04%</td>
<td>0.00%</td>
<td>0.00%</td>
</tr>
<tr>
<td>&lt; 1024</td>
<td>83.68%</td>
<td>73.73%</td>
<td>79.05%</td>
</tr>
<tr>
<td>&gt; 1024</td>
<td>16.32%</td>
<td>4.08%</td>
<td>20.95%</td>
</tr>
</tbody>
</table>
### Signature-based app. detection

- Port information offers no information for ports > 1024
- l7-filter system application signatures
- HTTP highly attractive for hiding other applications
- Most successful conns. trigger expected signature
- FTP higher percentage of false negatives

<table>
<thead>
<tr>
<th>Method</th>
<th>HTTP</th>
<th>IRC</th>
<th>FTP</th>
<th>SMTP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port (succ.)</td>
<td>93,429K</td>
<td>75,876</td>
<td>151,700</td>
<td>1,447K</td>
</tr>
<tr>
<td>Signature</td>
<td>94,326K</td>
<td>73,962</td>
<td>125,296</td>
<td>1,416K</td>
</tr>
<tr>
<td>expected port</td>
<td>92,228K</td>
<td>71,467</td>
<td>98,017</td>
<td>1,415K</td>
</tr>
<tr>
<td>other port</td>
<td>2,126K</td>
<td>2,495</td>
<td>27,279</td>
<td>265</td>
</tr>
</tbody>
</table>

### Signature detection: well known ports

- Some connections trigger more than one signature
- Not yet wide-spread abuse
- But some misappropriate use of well known ports

<table>
<thead>
<tr>
<th>Port</th>
<th>HTTP</th>
<th>IRC</th>
<th>SMTP</th>
<th>Other</th>
<th>No match</th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td>92,228,291</td>
<td>59</td>
<td>0</td>
<td>41,086</td>
<td>1,158,977</td>
</tr>
<tr>
<td>666x</td>
<td>1,217</td>
<td>71,650</td>
<td>0</td>
<td>4,238</td>
<td>524</td>
</tr>
<tr>
<td>25</td>
<td>459</td>
<td>2</td>
<td>1,415,428</td>
<td>195</td>
<td>31,889</td>
</tr>
</tbody>
</table>
Architecture for dynamic analysis

- Goals
  - Detection scheme independence
  - Dynamic analysis
  - Modularity
  - Efficiency
  - Customizability

- Design (USENIX Security'06)
  - Dynamic processing path
  - Per connection
dynamic analyzer trees

Bro: a flexible NIDS

- Facts
  - Open source
  - Developed since 1995 by Vern Paxson
  - Used in many research environments, e.g., UCB, LBL, TUM, The Grid, NERSC, ESnet, NCSA
  - Supports anomaly as well as misuse detection

- Design goals
  - Reliable detection of attacks
  - High-performance
  - Separation of base functionality from specific security policies
  - Robust against attacks on itself
Bro’s protocol analyzers

- Full analysis
  - HTTP, FTP, telnet, rlogin, rsh, RPC, DCE/RPC, DNS, Windows Domain Service, SMTP, IRC, POP3, NTP, ARP, ICMP, Finger, Ident, Gnutella

- Partial analysis
  - NFS, SMB, NCP, SSH, SSL, IPv6, TFTP, ...

- In progress
  - AIM, BGP, DHCP, Windows RPC, SMB, NetBIOS, NCP, Skype, BitTorrent

Reliable detection of non-standard ports

- UCB: 1 day internal remote
  - FTP servers: 6 17
  - HTTP servers: 568 54,830
  - IRC servers: 2 33
  - SMTP servers: 8 8

- MWN similar

- Non-standard port connection
  - UCB: 99% HTTP (28% Gnutella, 22% Apache)
  - MWN: 92% HTTP (21% BitTorrent, 20% Gnutella), 7% FTP
  - Two open HTTP proxy detected: now closed
  - SMTP server that allowed relay: now closed
Payload inspection of FTP data transfers

- FTP data transfers use arbitrary ports
- No longer a problem: dynamic prediction table

- File analyzer examines connection’s payload
  - Can determine file-type (LIBMAGIC)
  - Can check if actual file-type == expected file-type

- Extensions:
  - SMTP analyzer (using pipeline)
  - Virus checker

Detecting IRC-based Botnets

- Idea
  - Botnets like IRC protocol (remote control features)
  - Botnet detector on top of IRC analyser
    - Checks client nickname for typical patterns
    - Checks channel topics for typical botnet commands
    - Checks if new clients connect with IRC to identified bot-servers

- Results
  - MWN:
    - > 100 distinct IPs with Botnet clients
    - Now part of a automatic prevention system
  - UCB:
    - 15 distinct IPs
Summary: dynamic app. analysis

- **Ideas:**
  - Dynamic processing path
  - Per connection dynamic analyzer trees
- Operational at three large-scale networks
- Detected significant number of security incidents
- Bot-detection now automatically blocks IP

---

The Need for Collaboration between ISPs and P2P
P2P from an ISPs view

- **Good:**
  - P2P applications fill a void
  - P2P applications are easy to develop and deploy
  - P2P applications spur broadband demand

- **Bad:**
  - P2P systems form overlays at application layer
  - Routing layer functionality duplicated at app layer
  - P2P topology agnostic of underlay → performance loss
  - Traffic engineering difficult with P2P traffic

- ISPs are in a dilemma

---

ISP dilemma: Unstructured networks

Random/RTT-based peer selection → inefficient network resource usage
Solution? ISP-P2P cooperation

- Insight: ISP knows its network
  - Node: bandwidth, geographical location, service class
  - Routing: policy, OSPF/BGP metrics, distance to peers

Solution?: ISP-P2P cooperation

- Insight: ISP knows its network
  - Node: bandwidth, geographical location, service class
  - Routing: policy, OSPF/BGP metrics, distance to peers

- One proposal:
  - ISPs: offer oracle that provides network distance info
  - P2P: use oracle to build P2P neighborhoods
Solution?: ISP-P2P cooperation

- Insight: ISP knows its network
  - Node: bandwidth, geographical location, service class
  - Routing: policy, OSPF/BGP metrics, distance to peers

- One proposal:
  - ISPs: offer oracle that provides network distance info
  - P2P: use oracle to build P2P neighborhoods

- General proposal:
  - Offer network based interfaces to applications
  - To enable information exchange
  - To enable pushing services inside the network
  - Network based enablers...

Solution?: ISP-P2P cooperation

- Insight: ISP knows its network
  - Node: bandwidth, geographical location, service class
  - Routing: policy, OSPF/BGP metrics, distance to peers

- Oracle concept
  - Service of AS / ISP
  - Input: list of possible dst IPs
  - Output: ranked list of dst IPs
    - E.g. according to distances between src IP and dst IPs
Oracle service

Oracle-based peer selection → for topology and content exchange
Oracle service (3.)

Oracle-based peer selection → localizes topology and traffic

ISP-P2P cooperation?

- ISP-aided optimal P2P neighbour selection
  - Simple and general solution, open for all overlays
  - Run as Web server or UDP service at known location

- Benefits: P2P
  - No need to measure path characteristics
  - Easy to avoid bottlenecks => better performance

- Benefits: ISPs
  - Regains control over traffic
  - Cost savings
  - No legal issues (as no content is cached)
Evaluation

- Impact
  - Topology
  - Congestion
  - End-user performance

- Methodology
  - Sensitivity study
  - Use different ISP / P2P topologies
  - Use different user behavioral patterns
    - Content availability, churn, query patterns
  - Evaluate effects of on end-user experience
End-user performance evaluation

- Packet-level simulations
  - Scalable Simulation Framework (SSFNet)
  - Models for IP, TCP, HTTP, BGP, OSPF, etc.
  - Limited to about 700 overlay peers (memory constraints)

- Gnutella-based P2P system
  - Content search via flooding
  - Content exchange via HTTP

- Topologies: several
- User behavioral patterns: several

Topologies: ISP vs. P2P

- Germany
  - 12 ISP's (subset derived from published measurements)
  - 700 peers distributed according to ISP-published customer numbers

- USA
  - 25 Major ISP's (from Rocketfuel)
  - 700 peers distributed in AS's according to city population

- World topologies
  - Sub-sample of measured Internet AS-Topologies: 16 AS’s, 700 peers

<table>
<thead>
<tr>
<th>Tier</th>
<th>World1</th>
<th>World2</th>
<th>World3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tier1 (# AS / # peers)</td>
<td>1 / 10</td>
<td>1 / 355</td>
<td>1 / 50</td>
</tr>
<tr>
<td>Tier2 (# AS / # peers)</td>
<td>5 / 46</td>
<td>5 / 23</td>
<td>5 / 46</td>
</tr>
<tr>
<td>Tier3 (# AS / # peers)</td>
<td>10 / 46</td>
<td>10 / 23</td>
<td>10 / 42</td>
</tr>
</tbody>
</table>
P2P user behavior

- **Churn**: online/offline duration
  - Pareto and **Weibull** – close to observed behavior
  - Uniform – base comparison
  - Poisson – reflects worst-case scenario

- **Content**: type, availability and distribution
  - Constant size (512kB)
  - Pareto and **Weibull** – typical (many free-riders)
  - Uniform – base comparison
  - Poisson – hypothetical case (most peers sharing)

---

ISP experience: Intra-AS content

- **Content stays within ISPs network**
  - Without oracle 10 to 35%
  - With oracle 55 to 80%

- Consistent with Telefonica field trial results for BBC
ISP experience: Intra AS content (2.)

- Content stays within ISPs network

User experience: Download time

- Mean download time reduction: 1 – 3 secs (16 – 34%)
- Consistent across topologies
User experience: Download time (2.)

- Reduced mean download time

Overlay-underlay topology correlation

Random vs. biased P2P topology
Summary

❖ Oracle
  • Simple and easy to implement
❖ Evaluation shows
  • Overlay graph structure not affected
  • Reduced AS distance
    • P2P topology correlated with AS topology
  • Traffic congestion analysis
    • Reduces inter-AS traffic => load and costs
    • Traffic distribution close to theoretical optimum
❖ Benefits
  • ISPs: regain control of network traffic
  • P2P network: sees performance improvements

Potential advantage of Multi-Homing
Community network

Potential advantage of Multi-Homing

- **Idea**
  - Share broadband connections of private customers to third party users via WiFi
  - Enable nomadic Internet users to get access with better coverage at a lower cost

- **Advantage**
  - Public WiFi coverage will dramatically increase without rolling out costly infrastructure
  - More revenues are generated by nomadic users
  - Ubiquitous WiFi roaming can be achieved
Possible benefits of Multi-Homing?

- Explore impact of each component
  - Algorithm
  - Traffic
  - Network
    - DSL
    - Wireless

Traffic?

- Artificial
  - P2P BitTorrent
  - Web workload

- Real
  - Flow level traces
    - From TU-München: 2007
    - Crawdad
Algorithm?

- Direct
  - No rerouting
- FatPipe
  - Ideal case
- FullKnowledge
  - Min # of bandwidth limited flows
- MinLarge Flows
  - Min # of large flows

Approach

- Multifacet
  - Simulation
    - Fast special purpose simulator
    - Flow level
    - Fair sharing
    - Slow start
    - Fluid assumption
    - Different flow types
      - RTT limited
      - Interactive
      - Bandwidth limited
  - Test bed
Test bed

- Network: Wired and wireless
- Access: DSL or NistNet

Evaluation via simulator: 2Mbit DSL

- Significant benefit for bulky flows
Simulator vs. test bed?

- Good agreement

Simulator: direct vs. routed

- Clear difference :-(
Simulator: varying DSL connectivity

- Benefit increases with congestion

Simulator: varying DSL connectivity

- More congestion => more benefit
Lots of potential: heuristics are promising

Test bed: Bittorrent – NistNet

Three clients (1 Mbit) => factor 3 improvement :-(
Test bed: Web – 2Mbit NistNet

- Overhead for small flows (prototype) but significant benefits (~ factor 2.5 for flows > 0.5 sec)

Test bed: MWN – 2Mbit NistNet

- Overhead for small flows (prototype) but significant benefits (~ factor 3 for flows > 0.5 sec)
Test bed: MWN trace – 2Mbit DSL

- Mean improves by 2.2 for bulky (blue) flows
- Mean improves by 3 for bulky flows > 0.5 seconds

Test bed: MWN – DSL vs. NistNet

- Small differences
Test bed: MWN – wired vs. wireless

- Almost no difference

Benefit of flow-based routing

- It is possible (have prototype)
- Significant benefits (up to a factor of 3)
- Achievable benefit already quite nice. Still some room for improvement.

Methodology
- Simulation and test bed approach valuable
- Simulation: quick (and dirty)
- Test bed: slow but with real world constraints
Two approaches: Router vs. Client

Router:
- Operator-assisted/controlled
- Modifications required in the wireless router firmware, vendors participation
- No multihomed end user devices needed
- More accurate congestion information (wired/wireless)

Client:
- No operator control on client flow re-routing
- No modifications to the router, no involvement of vendors
- Only a software running in the client