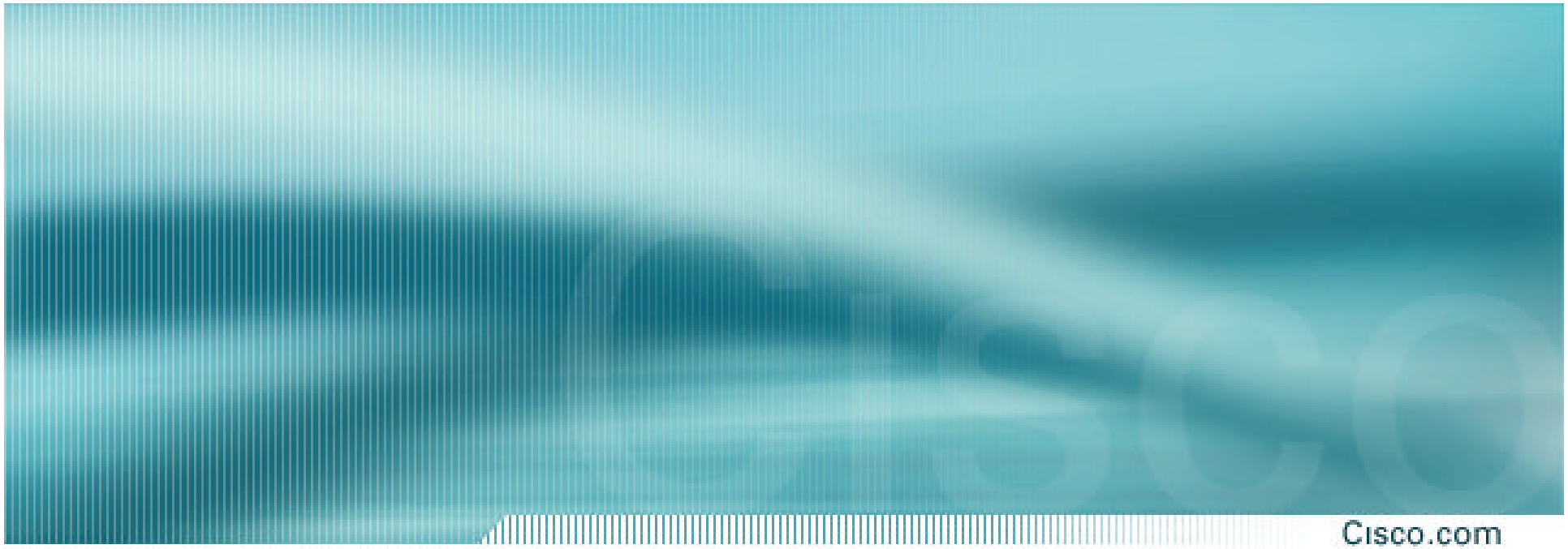


Peer to Peer Networking, the security perspective

Eric Vyncke

Cisco Systems

Distinguished Engineer

evyncke@cisco.com

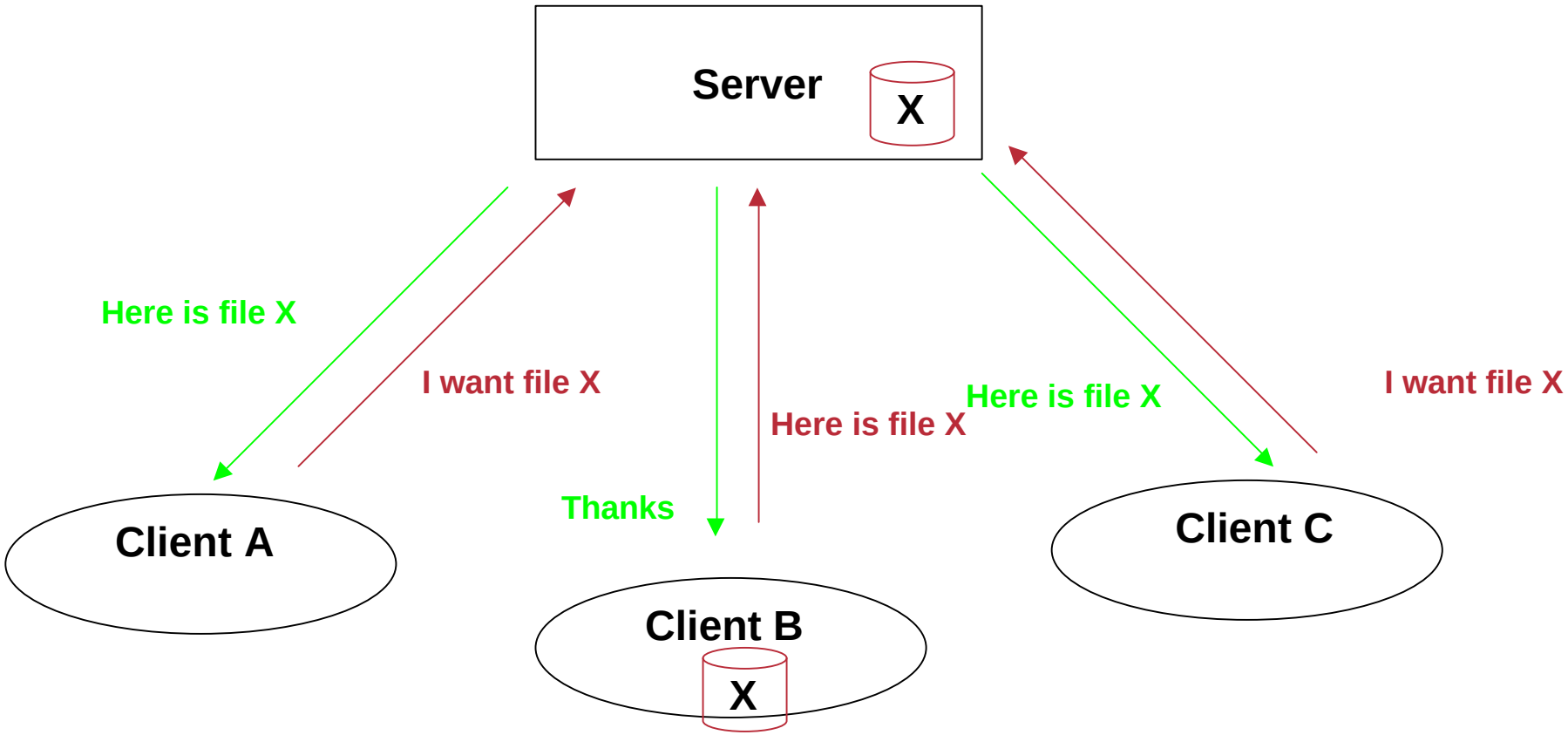
Agenda

Cisco.com

- Introduction to peer to peer networking
- Protocols description
- The threats
- Mitigation at the Desktop
- Blocking peer to peer networking

Usual information sharing

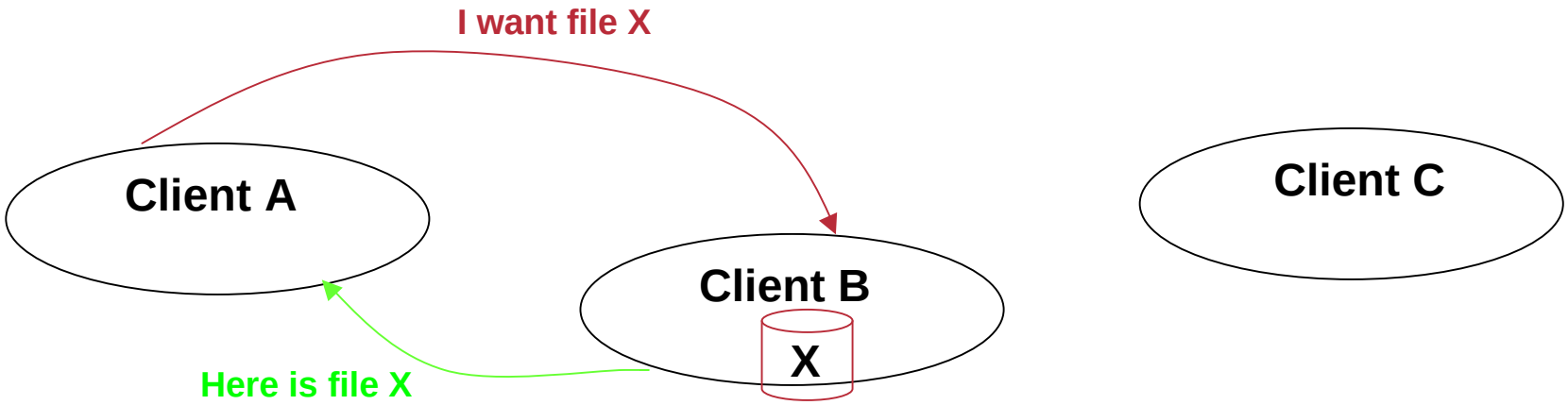
Centralized information repository



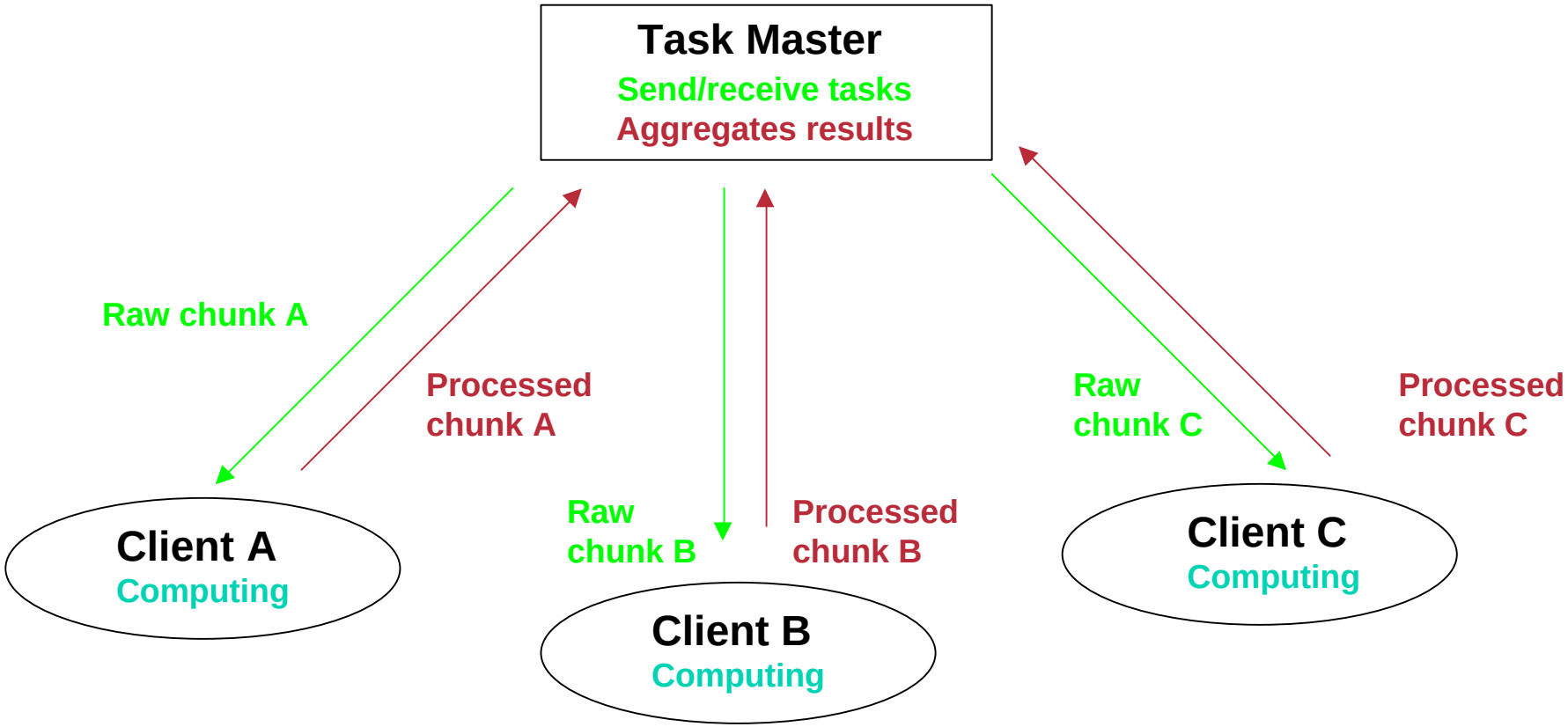
Information sharing in a peer to peer

Cisco.com

No centralized information repository



Cycle-Sharing (example: SETI@home)



Data is "vectorized" and sent to clients for ad hoc processing

Clients process in the background or when client is idle

Cycle Sharing: the Grid

- ***Named after the electrical power grid***
- **More than 10 years old**
- **Single application on geographically distributed, large parallel computers**
- **Security built-in**
  - **SSH, SSL, …**
  - **X.509 authentication, …**

Large technology companies are fully involved

- IBM (Global Grid Forum)
- Intel (P2P Forum)
- Sun (JXTA/JINI)
- Microsoft (.Net)
- Cisco is member of Global Grid and P2P

Who is writing P2P ?

- **Some are free**
- **Most contain spyware/adware**

  Part of the P2P code is used to display advertisments

  Some install browser plug-in to track the visited URL

  Remove with http://www.lavasoft.de (after removing the P2P does not work anymore ;-))

Agenda

Cisco.com

- Introduction to peer to peer networking
- Protocols description
- The threats
- Mitigation at the Desktop
- Blocking peer to peer networking

Architecture

- **Mosts of the protocols are using a three tiers architecture**

  Seed hosts **with static IP address/FQDN: give a couple of peers/servers dynamic addresses**

  Servers **with a file database: mainly for file search operation**

  Peers

Usual peer operations

Cisco.com

- **Seed**: connect to a well known host to learn about other peers
- **Register** node to the seed
- **Connect** to a server or peer
- **Publish** the locally shared files to the server
- **Search** on remote peers for a file
- **Retrieve** a file from multiple peers (to go faster!)

The most decentralized: Gnutella

- **The most interesting because it is the only peer to peer**

  Nodes are called either gnodes or servents (from server-cli**ent)**

- **Open specification**

  Extensions are in progress: IPv6, more security

  Projects to publish as IETF draft

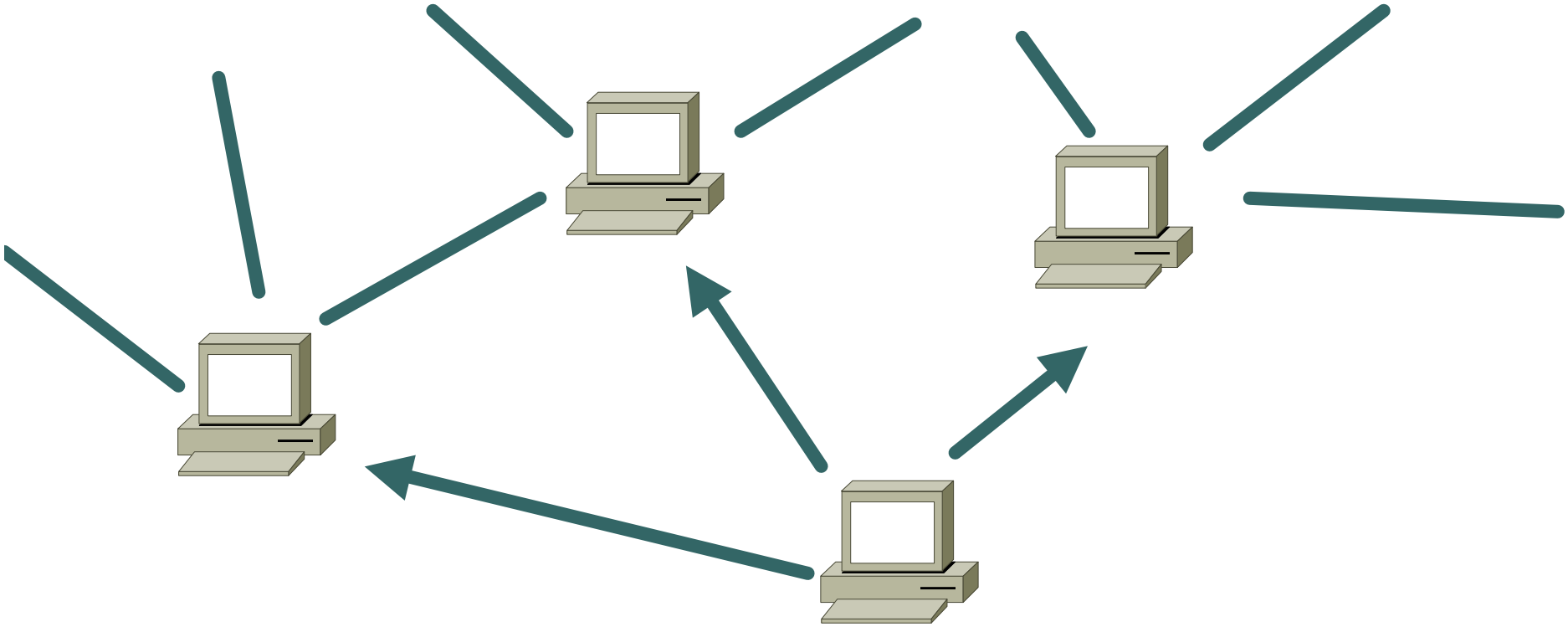
- **Multiple implementation: Limewire, Bearshare, Gnut, … on Windows, Linux, …**

Gnutella v0.4: Seed, Registration

- **Seed is either:**
  - **manual entry of other servents IP addresses**
  - **Or use of *gwebcache***
    - **•Used for both seed and registration**
    - **•Only gives 20 servents randomly**
    - **•Refreshed every 60 minutes**
    - **•Contains both servents IP addresses/TCP port and gwebcache URL**
- **Every node computes its own ClientID (randomly or based on Windows GUID)**

Gnutella: Connection to the network

- **Connection is a simple TCP connection to some peers** *(called servents – servers-clients)*



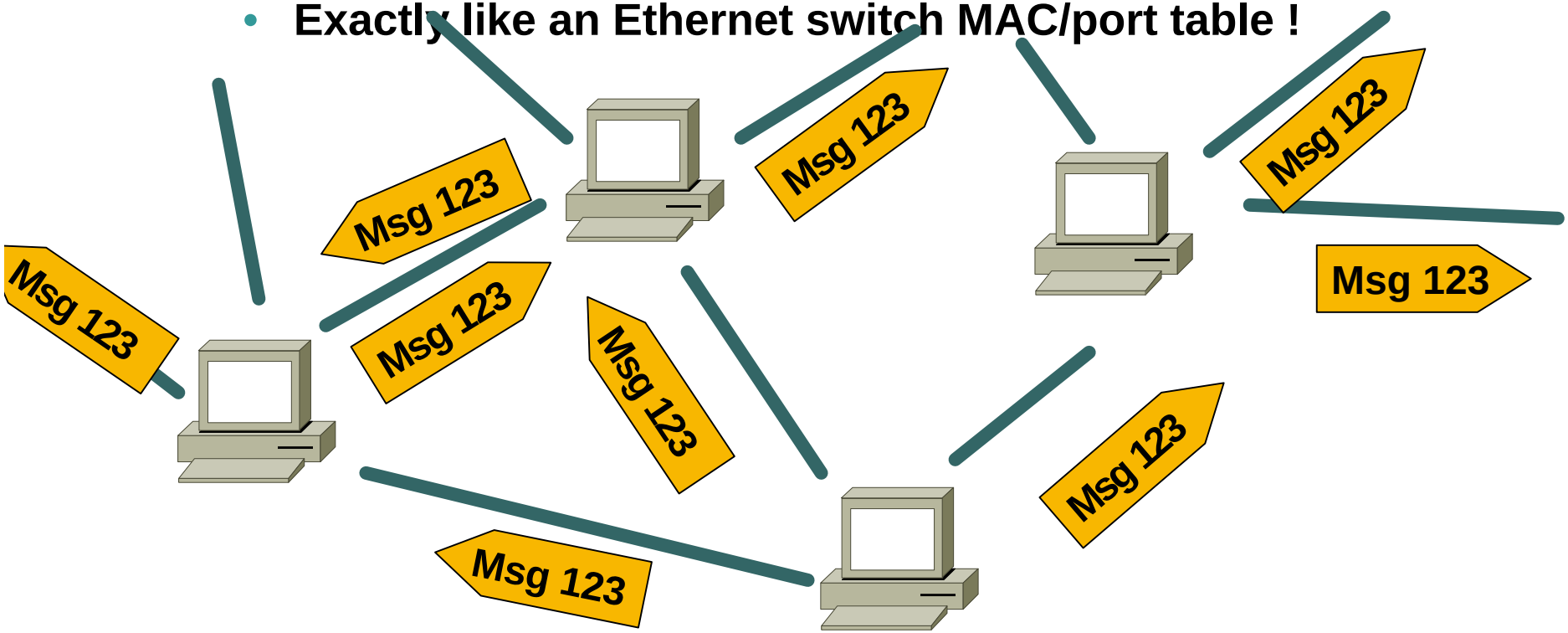
New servent with a unique & persistent ID

Gnutella messages

- **Only 5 messages are defined**
  - **PING: same as a broadcast ICMP ping**
  - **PONG: same as a ICMP ECHO_REPLY**
  - **QUERY: broacast a file search**
  - **QUERYHIT: response of a file search hit**
  - **PUSH: to bypass NAT/firewall**

Gnutella: communication is a learning bridge !

- Every message has a unique message ID
- Every servent maintains a table ID/TCP connection
- Packets with unknown message ID are flooded
- Exactly like an Ethernet switch MAC/port table !



Sending message ID 123

Gnutella: preventing loops

- **Two mechanisms are used**

  TTL: like in IP but any servent can also lower TTL if too high (typical value is 5)

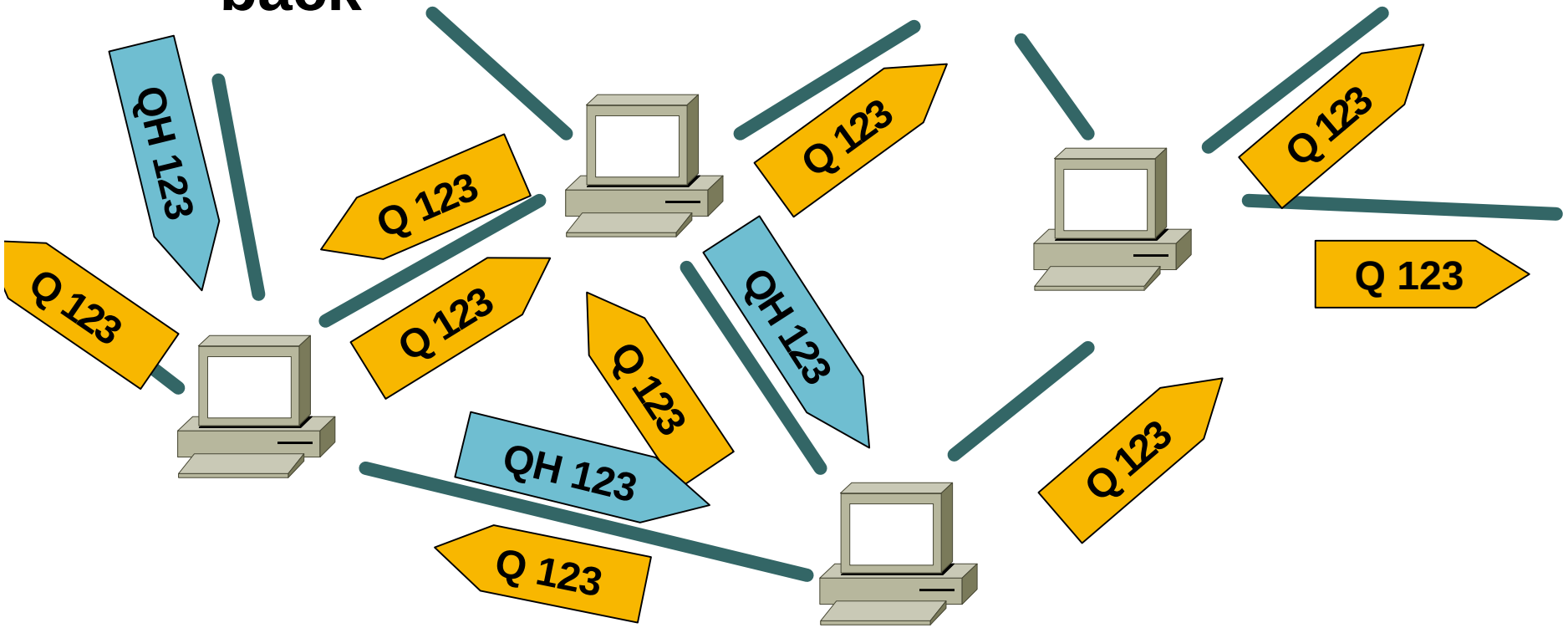
  Hop Count: always increasing (could also be used to remove looping packets)

- **BTW, privacy is enforced as there is neither source or destination ID in most messages**

  Only when hop count = 0, then the source is known

Gnutella: search queries and results

- Search queries are flooded until TTL is 0
- Only positive results, QueryHits, are sent back



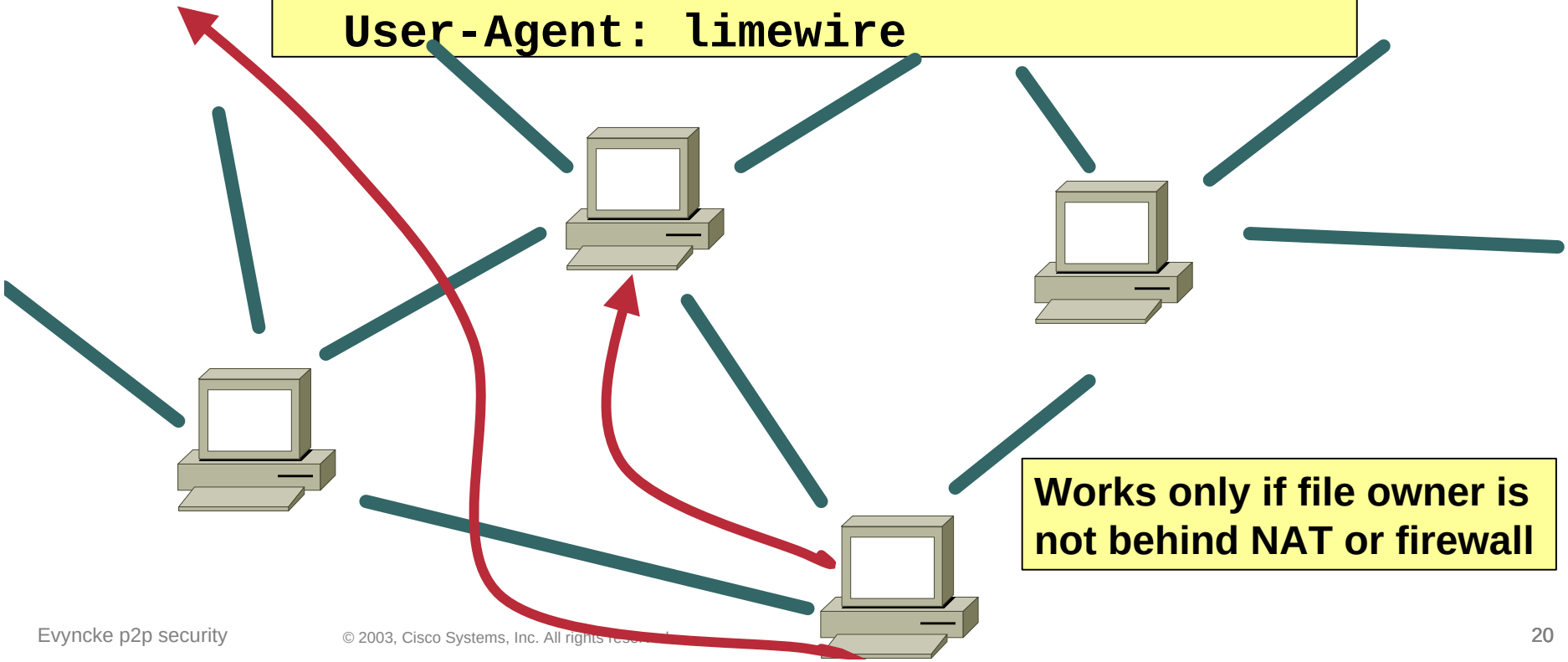
QueryHit

- **QueryHit replies contain more information**
  - **IP address, TCP port of the file owner**
  - **Servent ID of the file owner**

Gnutella: simple fetch

- ***Consumer*** **servent opens a HTTP connection to *remote* servent**

```
GET /get/123/Cisco_song.mp3 HTTP/1.0
Range: bytes=0-100000
User-Agent: limewire
```

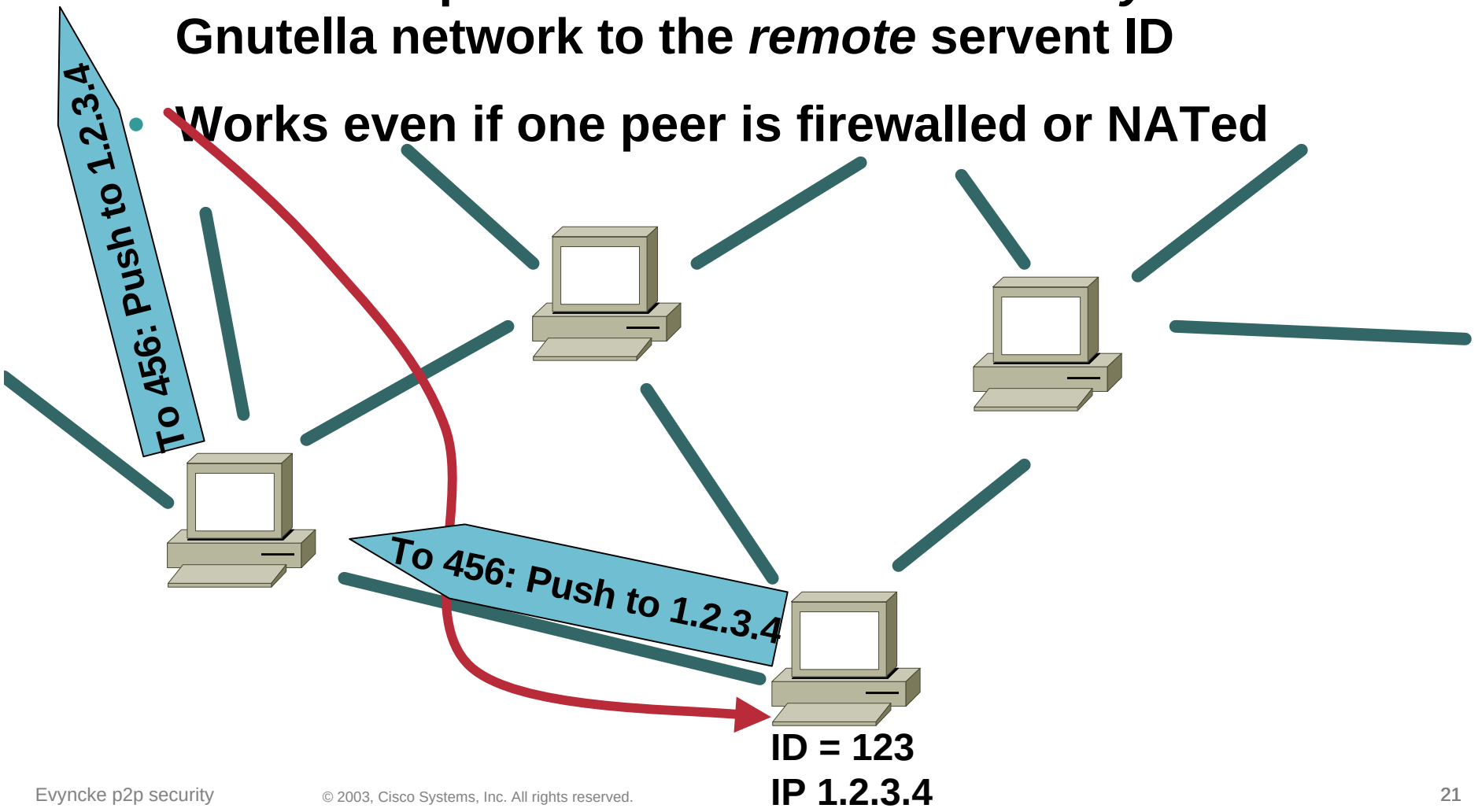


Works only if file owner is not behind NAT or firewall

Gnutella: when simple fetch fails

Cisco.com

- ***Consumer*** **peer sends a '*PUSH file to my IP*' on the Gnutella network to the *remote* servent ID**
- **Works even if one peer is firewalled or NATed**



Gnutella: next version 0.6

- Already deployed
- Concepts of two level of networks
  - Pure leaf servent: low bandwidth
    - Connect only to ultra-peers
    - Publish their files to ultra-peers
  - Ultra-Peer: high bandwidth
    - Connect to other ultra-peers & leaves
    - Execute queries in proxy of leaves

eDonkey: more centralized

- **Two kinds of peers:**

  Servers: store the list of files shared by clients

  Clients: with a single TCP connection to one server (to publish file and retrieve the server list) and multiple UDP 'connections' to all servers (for search)

- **The clients & servers are different programs**
- **The seed is purely manual**
- **Same concept of *push* as in Gnutella**
- **The UDP extended search is heavy as all clients are spraying search packets to all servers**

The proprietary one: Kazaa

- Protocol is proprietary ! And encrypted
- Protocol implementation is licensed by Sharmann Ltd (Netherlands – New Zealand)
- Also known as Fast Track
- Implementation: Morpheus (but has since diverged), KaZaa, …
- Two kinds of nodes: super-nodes and leaves

Agenda

Cisco.com

- Introduction to peer to peer networking
- Protocols description
- The threats
- Mitigation at the Desktop
- Blocking peer to peer networking

Content security: Sharing /1

- **Possible copyright violations**
  - **Music**
  - **Movie**
- **What about an organization liability if its network is used for P2P?**
  - **What about child pornography ?**

Content security: Sharing /2

- **Sharing more files than expected**

  Sharing the whole C:

  Bug in P2P client. E.g. Bearshare (Gnutella)

  http://xxx:6346/get/123/%5c..%5c..%5c..%5cwindows%5cwin.ini

  http://xxx:6346/get/123/%5c..%5c..%5c..%5cwindows%5cwin%2Eini

  Both retrieve C:\WINDOWS\WIN.INI without explicit sharing

Content Security: Retrieving /1

- **Retrieving executable content**
  - **Possible virus vector**
  - **Possible trojan/backdoor install**
- **Worse: P2P provides anonimoty on purpose => ideal vector**
- ***NB: also used as honey pots to attrack/log P2P users looking for illegal contents***

Content security: Retrieving /2

- **Retrieving executable content**

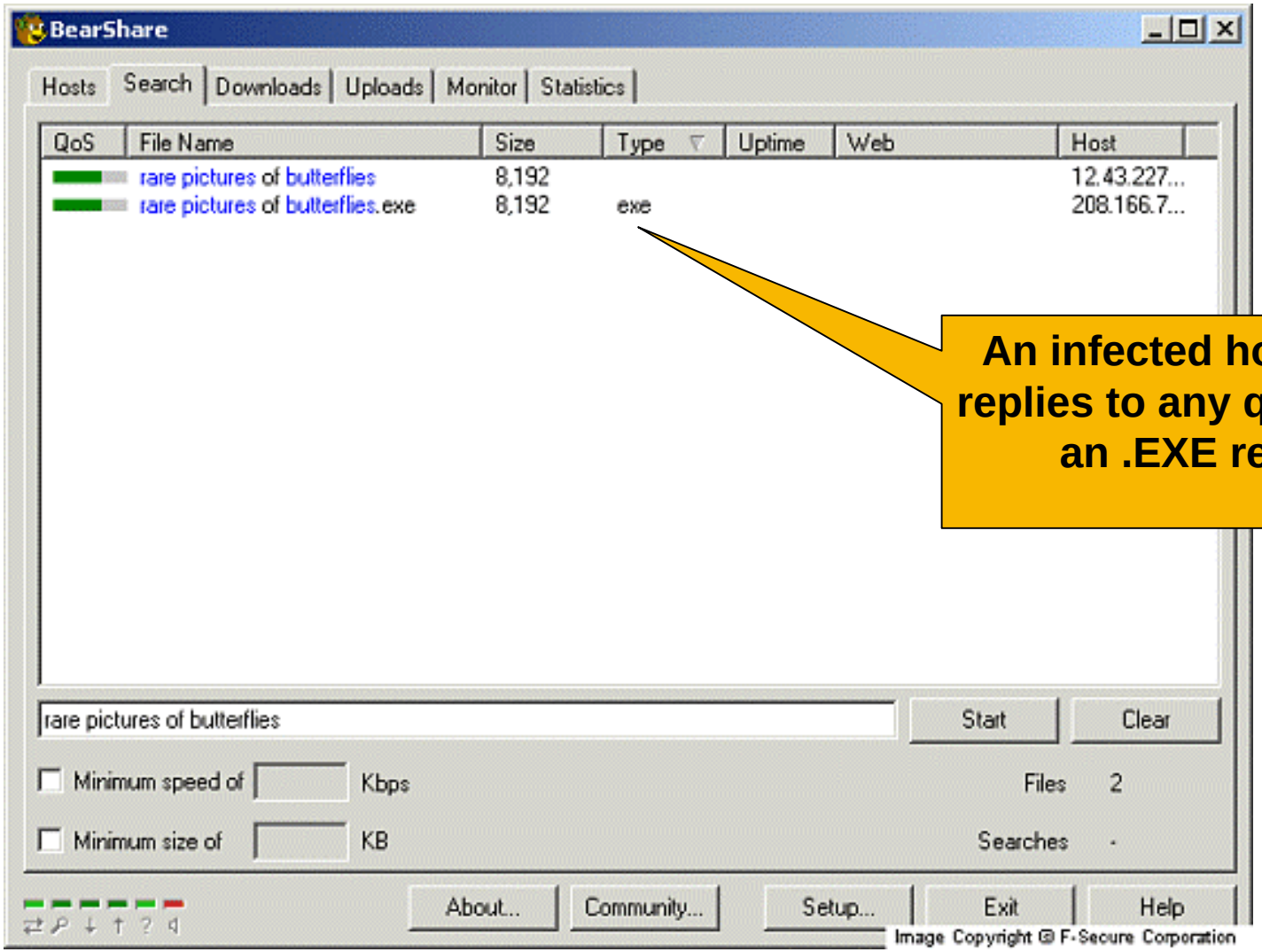
  Even .ASF or .MWF movie files can open URL… containing hostile code

  See http://www.pc-radio.com/gimp.mp3

Worms in P2P

- **Running worm will publish itself as:**
  - **Star_wars.exe**
  - **Quake_2.exe**
- **Naive users will download & execute**
- **Speading the worm further!**
- **See also:**
  - **Worm.Kazaa.Benjamin**
  - **Gnutella.Mandragore**

Gnutella.Mandragore



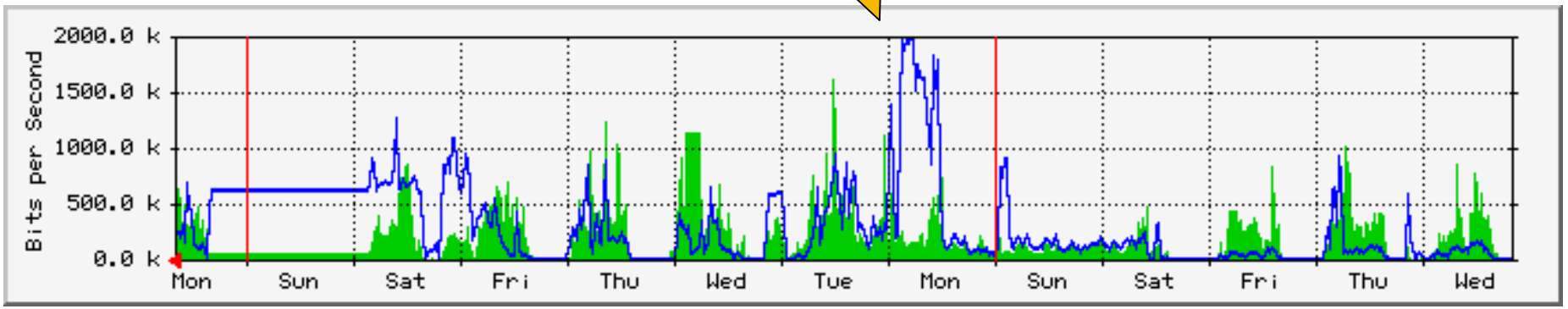
An infected host always replies to any queries with an .EXE result…

Bandwidth…

- **The most visible sign: upstream is utilized 100% even during nights**
- **Waste of money**
- **Can be too much for some non software IDS**
- **Content is random (compressed movie), can also trigger false positive on dumb IDS**

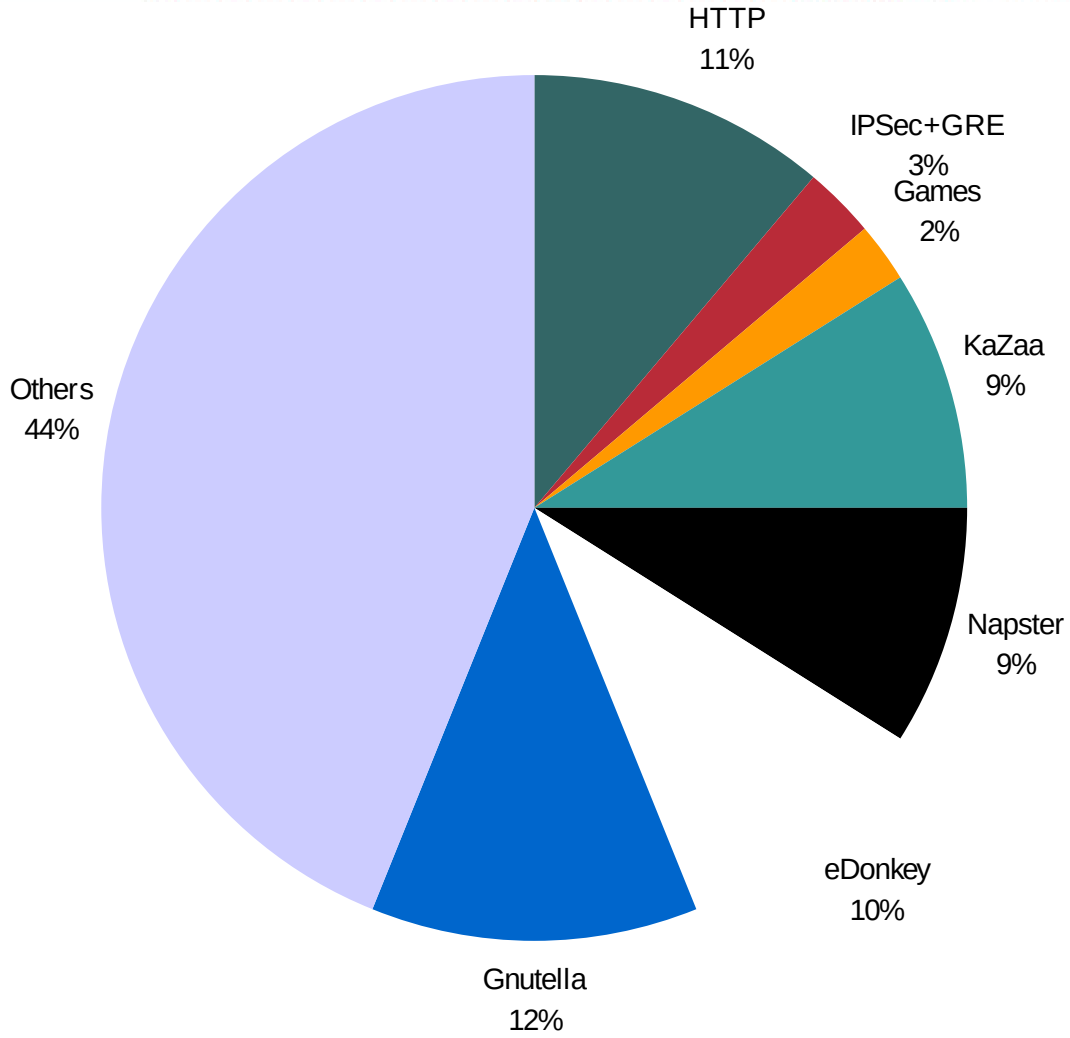
Real life example of bandwidth

Probably P2P:
-more upload than download
- unusual pattern
- night traffic



Data collected on a European NRN site in May 2002

Protocol Utilization on the Internet



P2P = 40%

Peering traffic data collected 9th of July 2002 in a large European ISP

Buffer overflow

- Peers are listening on a couple of UDP or TCP ports
- This may open a door to a buffer overflow attack
- Quite possible because a lot of code is not public
- Attractive target:
  - Million of peers
  - Always connected

Is this a Threat ?

- http://isc.sans.org/ on April 25, 2003

```
Top Attacked Ports
netbios-ns          137
Www                 80
ms-sql-m            1434
microsoft-ds        445
Smtp                25
Ident               113
---                 3136
netbios-ssn         139
gnutella-svc        6346
eDonkey2000         4662
```

Attack on the protocol

- **Most P2P protocols do not have authentication because authentication requires some form of centralization and trust…**
- **It should be easy to write a pseudo peer or server program**
  - **To bring part of the P2P network down**
  - **To log all search and push requests (for statistics, for caching, for logging or simply for law enforcement!)**

Attack on the P2P Network

- **As all peers are connected to multiple peers, the resiliency to failure is very high:**
  - **Gnutella can sustain 70% of nodes failure**
  - **Really needed because most peers stay connected for 1 hour**

Remote DoS

- **The ‘*push*’ operations could be diverted to launch DoS attack on a victim (but without an amplification) even behing a firewall**
- **Registrating victims as peers will force other peers to try to connect**
- **The eDonkey extended search could also be diverted to spray a host with 1000’s of UDP packets (with amplification)**

  It requires the hacker to publish the victim IP address as new server

Auto-update

- **Kazaa (and perhaps others) can use the P2P network in order to automatically install the next version**
- **This mechanism could be used to install thousands of Trojan…**

Adware & Spyware

- **The goal of those adware is to have the user to click on a banner**
  - **Yet another lost of time & ressource**
  - **Yet another door to hostile code on the clicked URL**
  - **Yet another piece of software to be trusted**

Agenda

Cisco.com

- Introduction to peer to peer networking
- Protocols description
- The threats
- Mitigation at the Desktop
- Blocking peer to peer networking

Mitigation at the Desktop

- **Most of the attacks**
  - **Worms & virus**
  - **Unexpected file access**
  - **Buffer overflow**
- **Can be detected by Host Intrusion Detection System**
  - **Actually an intrusion prevention**

Host Intrusion Detection System

- **Specific application hooking between kernel and other applications**
  - **Intercept all system calls**
  - **Check against rules**
    - **Valid file access?** C:\WINNT\SYSTEM32\CMD.EXE
    - **Valid network access?** Initiates TCP connections ?
  - **Can also detect buffer overflow**
- **=> Behavior based no need for signatures**
- **=> Efficient against known and unknown viruses, worms, …**

Agenda

Cisco.com

- Introduction to peer to peer networking
- Protocols description
- The threats
- Mitigation at the Desktop
- Blocking peer to peer networking

Block or throttle ?

- **Blocking**
  - **Attempt to block all P2P traffic**
  - **Is difficult: using well known ports, encrypted traffic, …**
  - **Can be bypassed: changing ports**
- **Throttle**
  - **Throttle the available bandwidth of P2P traffic in the network**
  - **As P2P is working, users will not try to evade**
    - **better logging**
    - **better forensics if required**

How to Recognize P2P ?

- **Currently**
  - **By the default TCP/UDP ports used**
  - **By the fixed seed/registation access**
- **Area of research**
  - **Use knowledge in network devices to detect P2P applications**
    - **Lot of long lasting flows**

Block Napster

- **Probably mostly unused nowadays**
- **Block TCP 6699/8888**

Blocking Gnutella

- **Difficult, Gnutella uses TCP 6346 per default but the port can be changed**
- **Seed is using multiple hosts**
- **=> not easy to block**

Blocking Kazaa

- **Kazaa uses UDP/TCP ports**
- **block all UDP/TCP traffic on port 1214 (default port but can also use port 80)**
- **Try to block the seed hosts: 213.248.112.0/24 (not tested)**

Blocking eDonkey2000

- **Seed: manual process initiated by web or IRC => difficult to block**
- **Traffic blocking: peers and servers can change the ports EXCEPT for the extended search**

  Peer to peer: TCP 4661

  Peer to server: TCP 4662

  Extended search: UDP 4665

Blocking AudioGalaxy

- Seed and registration based on a DNS lookup (for load balancing) then a query using TCP port 21 (the FTP port !) => difficult to block !
- Try to poison your DNS cache with wrong data like

  garlic.audiogalaxy.com 999999 IN A 127.0.0.1
- Transfer ports are TCP and dynamic in 41000-50000 => difficult to block !
- Try to block the seed hosts: 64.245.58.0/23 (not tested

Blocking iMesh

- Block TCP ports 5000 (register/search) and 4000-4999 (for transfer)
- Blocking port 5000 should be enough
- Try to block the seed hosts: 216.35.208.0/24 (not tested)

Blocking WinMX ?

- **Block UDP 6257 and TCP 6699 (but the user can change the ports…..)**
- **Try to block the seed hosts: 209.61.186.0/24 and 64.49.201.0/24 (not tested)**

Blocking the next one ?

- **Fighting against P2P is a never ending story…**
- ***Instead of blocking specific ports, rather permit only a couple of ports !***

Conclusion

- **P2P Networking:**
  - **Powerful concept (adhoc networking)**
  - **Currently misused**
  - **Existing and potential security vulnerabilities**
  - **Difficult but possible to mitigate at the network**
  - **Easier to mitigate at the desktop**
  - **User education !**

References

- http://testweb.oofle.com/filesharing/index.htm (more information on how to block)
- http://www.clip2.com/GnutellaProtocol04.pdf
- http://opennap.sourceforge.net/napster.txt
- http://homepage.mac.com/macdomeeu/dev/current/openag/agprotocol.html
- http://www.icir.org/vern/papers/cdc-usenix-sec02/index.html


CISCO SYSTEMS
EMPOWERING THE
INTERNET GENERATION