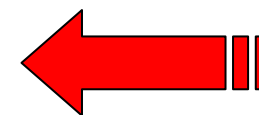




Evangelos Markatos  
FORTH-ICS  
[markatos@ics.forth.gr](mailto:markatos@ics.forth.gr)

<http://www.ics.forth.gr/~markatos>  
Institute of Computer Science (ICS)  
Foundation for Research and Technology – Hellas (FORTH)

- Motivation
  - What is network traffic monitoring?
  - Why is it important?
- Two methodologies for monitoring
  - **Active** network monitoring
    - Examples
  - **Passive** network monitoring
    - Examples
- pcap
- Introduce the rest of this tutorial
- Summary and Conclusions



Focus of  
this tutorial

- Committee on **Research Horizons in Networking** (formed in 2001)
  - **David Patterson, Chair**, University of California at Berkeley
    - RISC processors, RAID storage, NOWs – clusters of workstations
  - **David Clark**, MIT Laboratory for Computer Science
    - The “father” of the “end-to-end” argument, on top of which the Internet design is based
  - **Anna Karlin, Jim Kurose, Edward D. Lazowska, David Liddle, Derek McAuley, Vern Paxson, Stefan Savage, Ellen Zegura**
- The committee was assigned to
  - “**formulate a fresh look at networking research**”
- They prepared a report
  - “**Looking over the fence at networks: a Neighbor’s View of Networking Research**”
  - They identified three “**Grand Research Challenges**”



## The first GRAND Challenge in Computer Networking is to

*“... develop and deploy the technology to make it possible  
to record a day in the life of the Internet...”*



Committee on Research Horizons in Networking  
Clark, Lazowska, Patterson, Paxson, Savage, Zegura, ....  
2001



Evangelos Markatos [info@ist-lobster.org](mailto:info@ist-lobster.org)

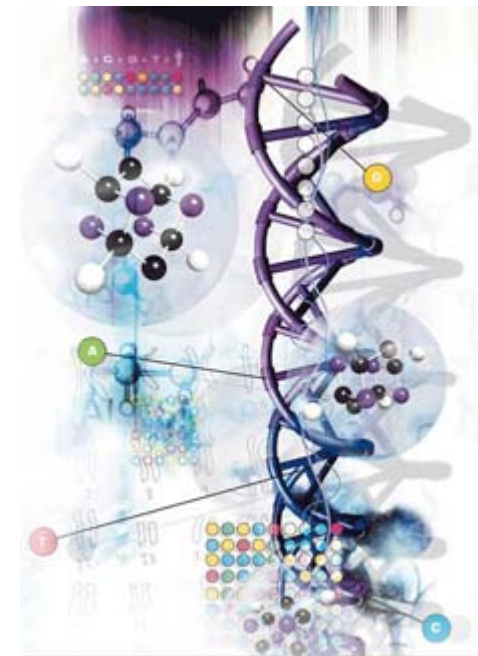




# The Challenge



- Why is it important?
  - “a data set with **typical days for the next 10 years of the Internet** might be a **treasure chest for networking researchers**”
  - **Measurement-based GRAND challenges**, such as the **human genome**, have served to
    - Crystallize research issues, and
    - Mobilize research efforts
  - Good network monitoring data are necessary for **operational needs**
    - Why is my network slow?
    - Which route do my packets follow?
    - Why is a particular flow missing lots of packets?
    - How much peer-to-peer traffic is there?
- **Next GRAND Challenge in Networking Research:**
  - **Monitor a day in the life of the Internet**



- So, do Network Traffic Monitoring
  - To get a better understanding on “what’s on the network”
- What can you do with monitoring?
  - Know the **state** of the Internet and your network
  - **Capacity** planning
  - **Traffic accounting**
    - Which application generates most traffic?
  - Understand the **performance** of individual applications
    - “why is my application so sloooooooow”?
  - Detect Security threats
    - DoS Attacks, Worm outbreaks





- Motivation
  - What is network traffic monitoring?
  - Why is it important?
- Two methodologies for monitoring
  - **Active** network monitoring
    - Examples
  - **Passive** network monitoring
    - Examples
- pcap
- Introduce the rest of this tutorial
- Summary and Conclusions



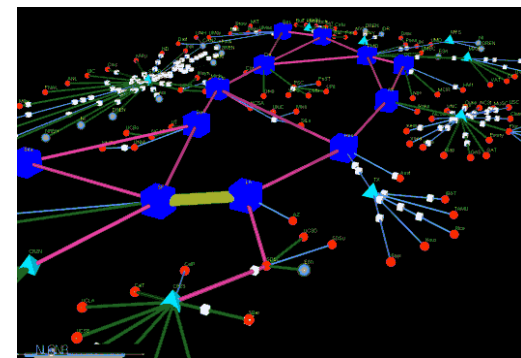
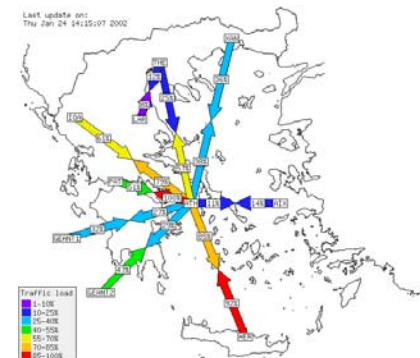
← Focus of  
this tutorial



# Active Monitoring



- Inject packets into the network
- Measure their arrival time, loss rate, etc.
- What can you do with it?
  - Measure **delay** (one-way / two-way)
  - Measure **bottleneck link bandwidth**
  - Find **network topology**
  - **What's up?** (in my network)
    - Which nodes are up and running?



```
%> ping www.ist-lobster.org
```

```
PING www.ist-lobster.org (192.87.30.11): 56 data bytes
```

```
64 bytes from 192.87.30.11: icmp_seq=0 ttl=49 time=308.7 ms
```

```
64 bytes from 192.87.30.11: icmp_seq=1 ttl=49 time=307.6 ms
```

```
64 bytes from 192.87.30.11: icmp_seq=2 ttl=49 time=244.4 ms
```

```
--- www.ist-lobster.org ping statistics ---
```

```
3 packets transmitted, 3 packets received, 0% packet loss
```

```
round-trip min/avg/max = 244.4/286.9/308.7 ms
```

```
%>
```

# Traceroute: find all intermediate routers between a source and a destination computer

```
%> traceroute www.ist-lobster.org (from Crete)
```

```
traceroute to www.ist-lobster.org (192.87.30.11), 30 hops max, 40 byte packets
```

```
 1 147.52.17.1 (147.52.17.1) 1.050 ms 0.690 ms 0.592 ms
 2 olympos-e43.lanh.uoc.gr (147.52.12.1) 1.626 ms 1.033 ms 0.840 ms
 3 heraklio-uch-ATM.grnet.gr (194.177.209.141) 129.528 ms 112.644 ms 123.465 ms
 4 heraklio2-to-heraklio.backbone.grnet.gr (194.177.209.77) 124.791 ms 116.749 ms 119.965 ms
 5 Syros-to-Heraklio2.backbone.grnet.gr (195.251.27.81) 136.089 ms 104.469 ms 81.116 ms
 6 athens3-to-Syros.backbone.grnet.gr (195.251.27.10) 72.664 ms 62.814 ms 67.341 ms
 7 grnet.gr1.gr.geant.net (62.40.103.57) 81.392 ms 102.067 ms 79.488 ms
 8 gr.de2.de.geant.net (62.40.96.82) 129.641 ms 134.589 ms 144.765 ms
 9 de2-2.de1.de.geant.net (62.40.96.54) 139.478 ms 158.336 ms 146.815 ms
10 de.nl1.nl.geant.net (62.40.96.102) 180.696 ms 162.904 ms 173.813 ms
11 surfnet-gw.nl1.nl.geant.net (62.40.103.98) 184.078 ms 158.921 ms 160.933 ms
12 PO11-0.CR1.Amsterdam1.surf.net (145.145.166.33) 145.367 ms 150.166 ms 142.117 ms
13 PO0-0.AR5.Amsterdam1.surf.net (145.145.162.2) 163.605 ms 144.161 ms 177.526 ms
14 145.145.18.46 (145.145.18.46) 178.350 ms 175.365 ms 166.334 ms
15 * * 145.145.18.46 (145.145.18.46) 176.079 ms !X
16 * 145.145.18.46 (145.145.18.46) 171.861 ms !X *
17 145.145.18.46 (145.145.18.46) 192.753 ms !X * 180.104 ms !X
```

VisualRoute 5.0b

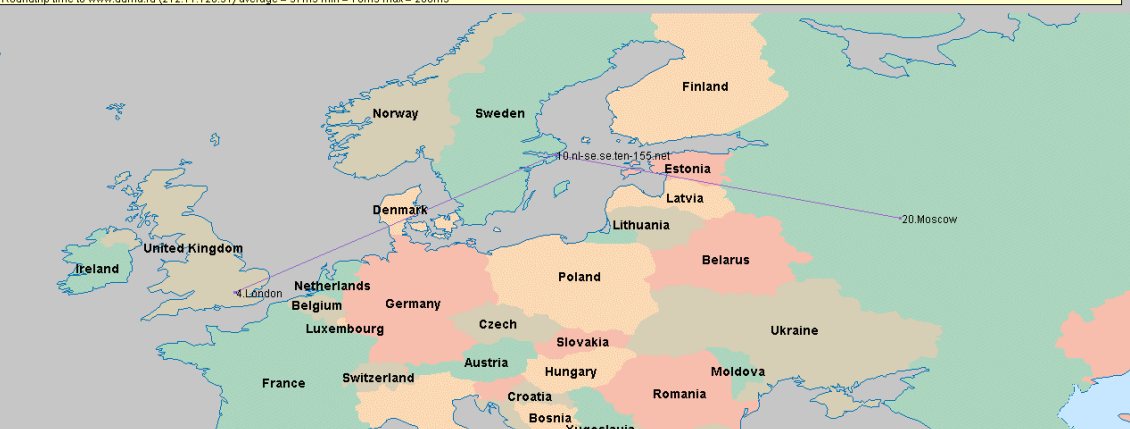
Host: <http://www.duma.ru> IP Addresses: **212.11.128.31** Recent Hosts

**Report for www.duma.ru [212.11.128.31]**


Analysis: Node 'www.duma.ru' was found in 22 hops (TTL=107). It is a HTTP server (running Novell-HTTP-Server3.1R1).

Hop	% Loss	IP Address	Node Name	Location	Timezone	rms	Graph	Network
0		128.40.59.193	wolf.casa.ucl.ac.uk	-	-	-	0	321 University College London
1		128.40.59.245	cisco-2.bartucl.ac.uk	-	-	0	0	University College London (private use)
2		10.0.121.45	-	-	-	0	1	University College London
3		128.40.255.153	-	-	-	1	1	The London MAN
4		194.83.100.62	atmr-ulcc.lmn.net.uk	London, UK	0.0	1	1	University of London Computer Centre
5		146.97.40.65	gl0-0-0-ext-gw6.ja.net	-	-	1	1	University of London Computer Centre
6		128.96.1.15	ten155-gw.ja.net	-	-	9	9	IP allocations for TEN-155 ATM PVC
7		212.1.192.149	janet.uk.ten-155.net	51.50 N, 0.11 W (United Kingdom)	0.0	0	0	IP allocations for TEN-155 PoP equipment
8		212.1.193.154	ge.uk40.ten-155.net	-	-	10	10	212.1.197.57
9		212.1.197.57	nl-uk-1.nl40.ten-155.net	-	-	40	40	IP allocations for TEN-155 ATM PVC
10		212.1.192.102	nl-se.se.ten-155.net	59.33 N, 18.05 E (United Kingdom)	0.0	63	63	IP allocations for TEN-155 external peerings
11		212.1.194.26	stockholm5.se.eqip.net	-	-	+1.0	42	Stockholm Interconnect
12		194.88.129.25	Stockholm-DJK.ebone.net	Stockholm, Sweden	+1.0	46	46	Ebone backbone 3
13		195.158.226.77	sesto502-lb-p0-0.ebone.net	-	-	+1.0	47	Ebone backbone 3
14		195.158.226.54	-	(Sweden)	-	81	81	Sovam Teleport
15	10	194.186.157.181	cisco0.Moscow.ST.NET	Moscow, Russia	-	70	70	Sovam Teleport
16	20	194.186.157.182	cisco02.Moscow.ST.NET	Moscow, Russia	-	85	85	Sovam Teleport
17	10	194.67.16.219	ccr-1.Moscow.ST.NET	Moscow, Russia	-	88	88	Sovam Teleport
18	10	194.186.0.199	tr-pop-gw.Moscow.ST.NET	Moscow, Russia	-	84	84	Glas-Internet Ltd
19		195.218.254.83	MOS-gw.glas.net	(Russia)	-	80	80	Moscow Mayor's Office
20	10	212.11.128.31	duma.ru	Moscow, Russia	-	80	80	Moscow Mayor's Office
21	10	212.11.128.31	duma.ru	Moscow, Russia	-	97	97	Moscow Mayor's Office
22		212.11.128.31	www.duma.ru	Moscow, Russia	-	-	-	Moscow Mayor's Office

VisualRoute Report for www.duma.ru produced at 18:55 on 24 November, 2000.  
Roundtrip time to www.duma.ru (212.11.128.31) average = 97ms min = 70ms max = 200ms



File Database Help



Nr	Hostname	IP number
10	sl-bp021-stk-1-2.sprintlink.net	144.252.4.70
11	sl-ucberkeley-1-1-0-T3.sprintlink.net	144.228.146.50
12	f5-0.inr-666-eva.berkeley.edu	198.128.16.21
13	f1-0-0.inr-107-eva.Berkeley.EDU	128.32.2.1
14	f8-0.inr-100-eva.Berkeley.EDU	128.32.235.100
15	amber.Berkeley.EDU	128.32.25.12

Info



- Motivation
  - What is network traffic monitoring?
  - Why is it important?
- Two methodologies for monitoring
  - **Active** network monitoring
    - Examples
  - **Passive** network monitoring
    - Examples
- pcap
- Introduce the rest of this tutorial
- Summary and Conclusions



 **Focus of  
this tutorial**



# Passive Traffic Monitoring



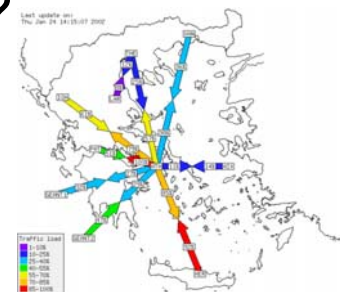
- What is it?
- Non-intrusive traffic monitoring
  - Much like a **telescope**
  - **Does not inject packets** in the network
- It **passively captures** information from passing packets such as
  - High-level network flows (CISCO Netflow)
  - Network packet headers (NLANR)
  - Entire network packets (incl. payload)
    - if allowed
    - maybe stripped/anonymized (to be shared with a broader audience)



# Passive Monitoring: What can it be used for? Performance

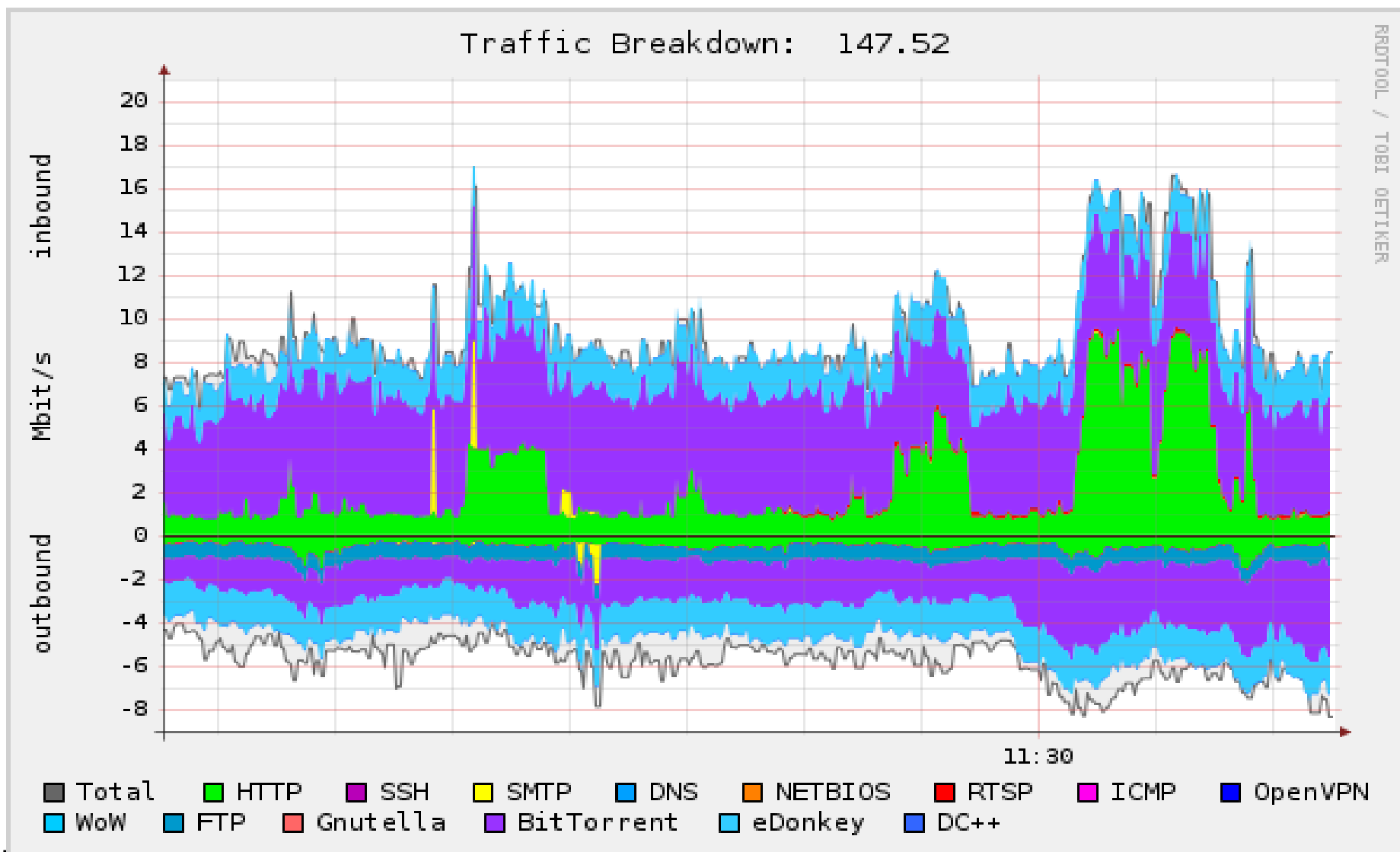


- Traffic Categorization/Accounting:
  - What % of my traffic is due to email?
  - Which subnet generates most outgoing traffic?
- Bandwidth Estimation
  - What % of my bandwidth is available now?
  - What % of my bandwidth is being used?
- Study trends:
  - How does the application mix in the traffic changes with time?
    - ftp in the 80's, www in the 90's, p2p in the 00's
  - How does peer-to-peer traffic changes with time?
- Performance Debugging of individual applications
  - Why is **my** application so sloooow?





# Example Application: Traffic Categorization



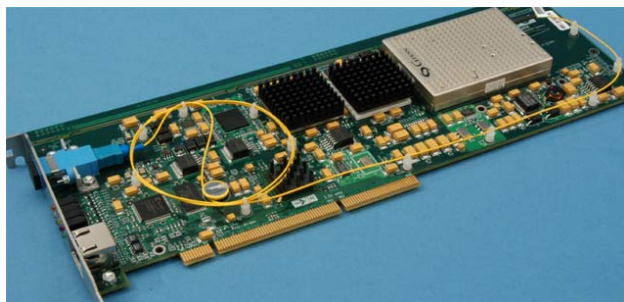
RRDTOOL / TOBI OETIKER

# Passive Monitoring: What can it be used for? Security

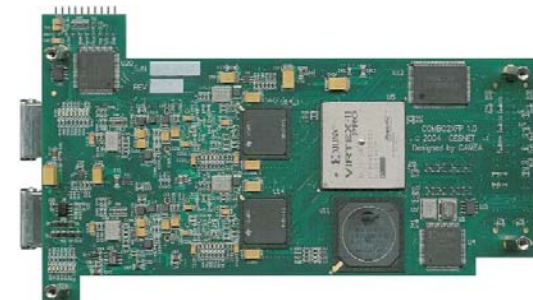
- **Intrusion Detection**
  - Are any of my computers compromised?
    - Do they participate in a botnet?
  - Is there any attacker trying to intrude into my network?
- **Large-scale Attack Detection** – Detection of Epidemics
  - DoS Attack detection
    - e.g. Detect sharp increases in TCP/SYN packets
  - Zero-day worm detection
    - e.g. Detect lots of identical packets, never seen before, from several sources to several destinations
    - e.g. Detect worm characteristics
      - such as NOP sleds: long sequences of executable code
- **Network Telescopes**
  - They monitor unused IP addresses (“dark matter”)
  - Ordinarily, unused IP addresses should not receive traffic
  - Observe victims of DoS attacks
    - “back-scatter” traffic, “background radiation”
  - Observe infected hosts
  - Port scans



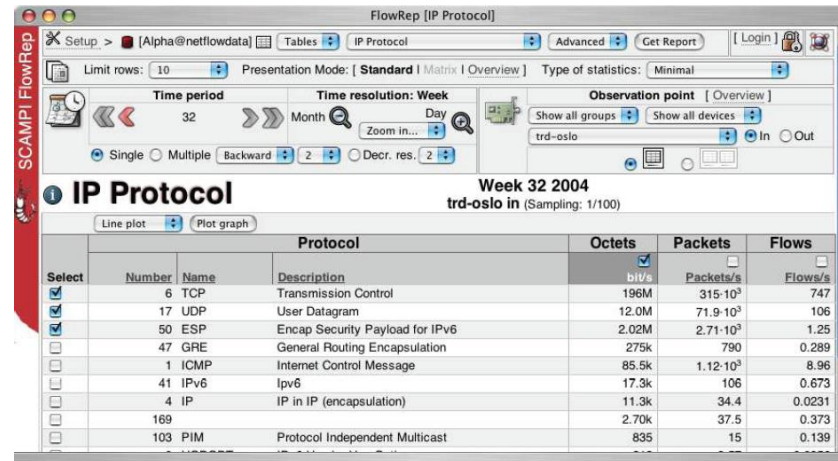
- Equipment varies from
  - low-end (for low-speed networks) to
  - Sophisticated equipment (for high-speed networks)
- Low-end passive monitors (100Mbps – 1Gbps)
  - An ordinary PC
  - An ordinary network Interface (i.e. an Ethernet card)
    - put in promiscuous mode
  - Mirror all packets from a router to a port connected to the above PC



- High-end passive monitors (1Gbps – 10Gbps)
  - High-end computer
  - Specialized network interface
    - DAG Cards (Endace)
    - Combo cards (SCAMPI project)
    - Hardware-based filtering capabilities
      - Process packets at line speeds



- pcap: packet capture library from Berkeley
- MAPI: Monitoring API (Application Programming Interface)
  - <http://www.ist-lobster.org/downloads>
  - Developed within the IST SCAMPI project
    - co-funded by EU
- Net-flow-related tools
  - Graphical interfaces



The screenshot shows the FlowRep [IP Protocol] web interface. The main content is a table titled "IP Protocol" for "Week 32 2004" and "trd-oslo in (Sampling: 1/100)". The table has columns for "Protocol", "Octets", "Packets", and "Flows". The "Octets" column is checked, and the "Packets" column is unchecked. The table lists various protocols with their respective statistics.

Select	Number	Name	Description	Octets bit/s	Packets Packets/s	Flows Flows/s
<input checked="" type="checkbox"/>	6	TCP	Transmission Control	196M	315·10 <sup>3</sup>	747
<input checked="" type="checkbox"/>	17	UDP	User Datagram	12.0M	71.9·10 <sup>3</sup>	106
<input checked="" type="checkbox"/>	50	ESP	Encap Security Payload for IPv6	2.02M	2.71·10 <sup>3</sup>	1.25
<input type="checkbox"/>	47	GRE	General Routing Encapsulation	275k	790	0.289
<input type="checkbox"/>	1	ICMP	Internet Control Message	85.5k	1.12·10 <sup>3</sup>	8.96
<input type="checkbox"/>	41	IPv6	IPv6	17.3k	106	0.673
<input type="checkbox"/>	4	IP	IP in IP (encapsulation)	11.3k	34.4	0.0231
<input type="checkbox"/>	169			2.70k	37.5	0.373
<input type="checkbox"/>	103	PIM	Protocol Independent Multicast	835	15	0.139



- Motivation
  - What is network traffic monitoring?
  - Why is it important?
- Two methodologies for monitoring
  - **Active** network monitoring
    - Examples
  - **Passive** network monitoring
    - Examples
- **pcap**
- Introduce the rest of this tutorial
- Summary and Conclusions



← Focus of  
this tutorial



- Packet capture Library
- Developed to capture packets
  - And dump them to disk
- Basic Steps:
  - Open a network interface
  - Get one packet at a time
  - Print it – dump it to the disk



- Example: Grab one packet:

```
main() {  
    /* open a network interface */  
    descr = pcap_open_live(dev, BUFSIZ, 0, 1, errbuf);  
    /* Get next packet */  
    packet = pcap_next(descr, &hdr);  
    /* print its length */  
    printf("Grabbed packet of length %d\n", hdr.len);  
}
```



- Example: Grab several packets:

```
main() {  
    /* open a network interface */  
    descr = pcap_open_live(dev, BUFSIZ, 0, 1, errbuf);  
    while (1) {  
        /* Grab packets for ever */  
        packet = pcap_next(descr, &hdr);  
        /* print its length */  
        printf("Grabbed packet of length %d\n", hdr.len);  
    }  
}
```



# pcap (several packets)



- Example: Grab several packets:

```
main() {
/* open a network interface */
descr = pcap_open_live(dev,BUFSIZ,0, 1,errbuf);
pcap_loop (descr,atoi(argv[1]),my_callback, NULL);
}

/*callback function */
void my_callback (u_char *u,const struct pcap_pkthdr* pkthdr,
const u_char* packet)
{
/* just count the number of packets */
static int count = 1;
printf("%d, \n",count);
count++;
}
Grab 100 packets:
%>a.out 100
```



# pcacp (with filters)



- Example: Suppose that you want to capture only packets destined to your web servers (**destination port 80**):

```
main() {
descr = pcap_open_live(dev, BUFSIZ, 0, 1, errbuf);
/* install a filter */
pcap_compile(descr, &fp, "dst port 80", 0, netp)
pcap_setfilter(descr, &fp)

pcap_loop (descr, atoi(argv[1]), my_callback, NULL);
}

/*callback function */
void my_callback (u_char *u, const struct pcap_pkthdr* pkthdr, const
u_char* packet)
{
static int count = 1;
printf("%d, \n", count);
count++;
}
```

- Excellent for capturing and dumping packets by a single application
- Several applications
  - High overhead – copies all packets to all applications
- Limited functionality
  - No string searching, no packet counting, etc.



# Interested in pcap?



- <http://www.cet.nau.edu/~mc8/Socket/Tutorials/section1.html>
- <http://www.tcpdump.org/>
- <http://www.winpcap.org/>

- MAPI
  - Daemon-based packet monitoring
  - Rich functionality (string searching, counters, anonymization, etc.)
- FFPF (Fairly Fast Packet Filters)
- Flow-level passive monitoring
  - General introduction
    - NetFlow and IPFIX
  - NetFlow-based applications
    - Stager (UNINETT)
    - NERD (TNO)
- Ruler Anonymization Language



## *“Monitor a Day in the Life of the Internet”*

Committee on Res. Horizons in Networking



- Traffic Monitoring help us understand what's on the network
- Passive Network Traffic Monitoring applications for :
  - Performance
    - traffic accounting/categorization
    - Performance debugging
  - Security
    - DoS attack detection,
    - Internet epidemics
    - Intrusion Detection



Evangelos Markatos  
FORTH-ICS  
[markatos@ics.forth.gr](mailto:markatos@ics.forth.gr)

<http://www.ics.forth.gr/~markatos>  
Institute of Computer Science (ICS)  
Foundation for Research and Technology – Hellas (FORTH)