Mehr Klarheit für Ihre Netzwerkperformance

Die Integration von ntop in die Überwachungslösung NetEye

by Georg Kostner

About Würth Phoenix

- IT and Consulting Company of the Würth-Group
- Headquarter in Italy, European-wide presence, more than 100 employees
- International experience in Business Software and IT Management
- System Monitoring – Network Monitoring
- ITIL certified, Nagios Solution Provider, OTRS Certified Partner

Facts & figures

- More than 600 customers worldwide
- Over 7.000 ERP and CRM users
- 25.000 monitored hosts
- 4 offices in 3 countries
- HQ in Italy
- Core offers in Business Software and IT System Management

Our mission is to improve the business productivity of our customers by managing working processes more efficiently.
To assure this we offer complete and international proven IT- solutions in a well-known Würth-quality.

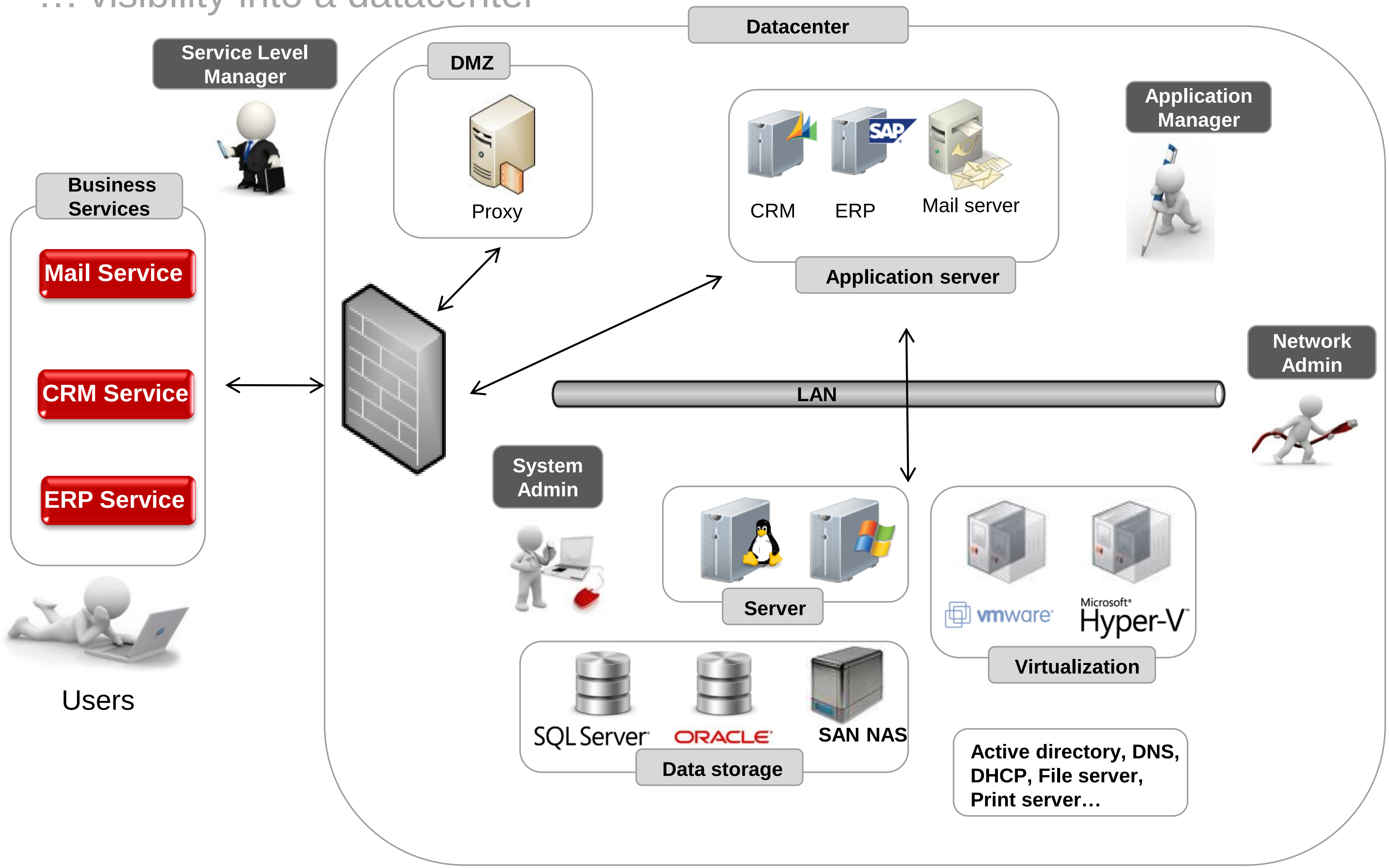
WÜRTHPHOENIX NetEye

…the market proven alternative

- WÜRTHPHOENIX NetEye is an **Open Source package** to monitor the IT infrastructure
- The solution has been developed to simplify your IT infrastructure management increasing its reliability
- NetEye is based on proven Open Source monitoring solutions with over 250.000 estimated worldwide users
- Würth Phoenix has **10 years experience** in implementing monitoring system and provides support services.
- Würth Phoenix is **Nagios Solution Provider**

Network Monitoring with NetEye

… visibility into a datacenter



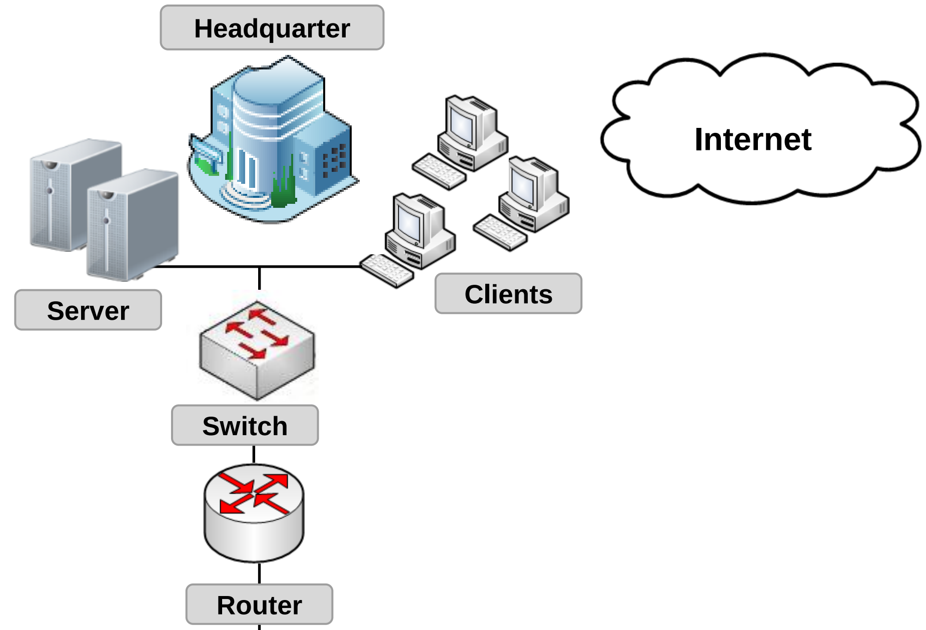
Network Traffic Analysis with NetFlow

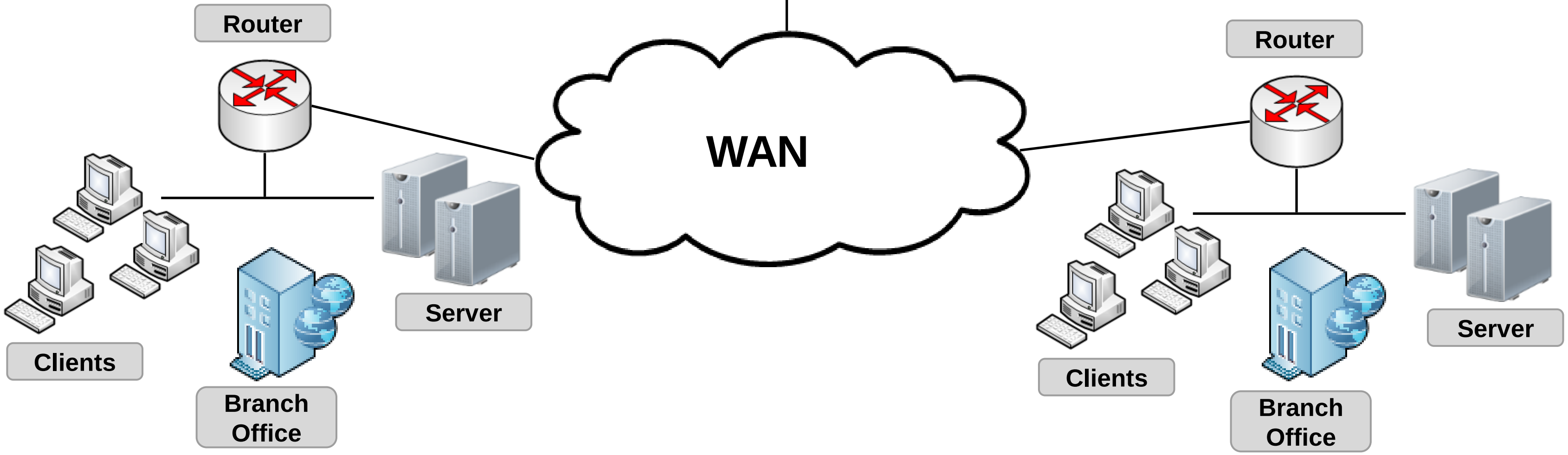
…measure your latency, bandwidth, in-outbound traffic

Network Monitoring

WÜRTHPHOENIX NetEye

Latency
Bandwidth
In/outbound Traffic





Network Monitoring

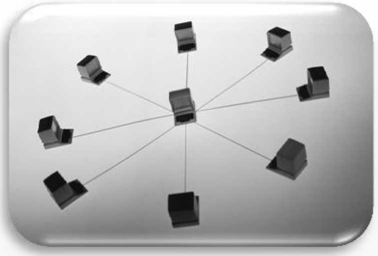
…what you can analyze with NetEye

- Network latency and bandwidth monitoring point to point, network interface in/outbound

- Definition of active/passive checks (SNMP Requests, SNMP Traps)

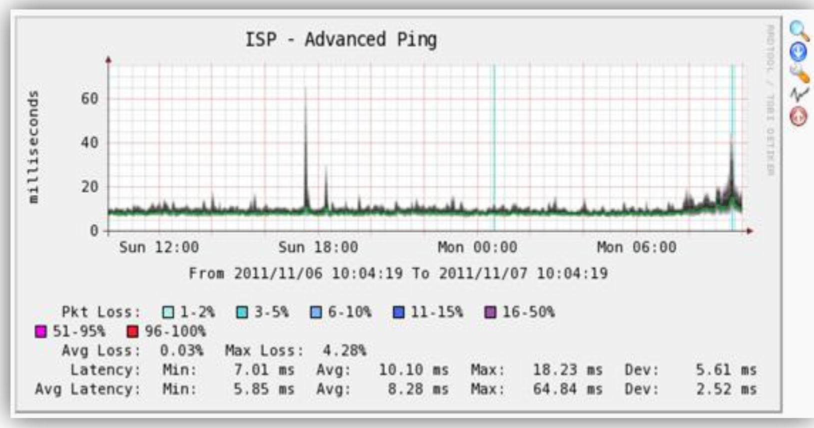
- Graphs for in/outbound traffic min, avg, max values on switch, routers

Analyze your network

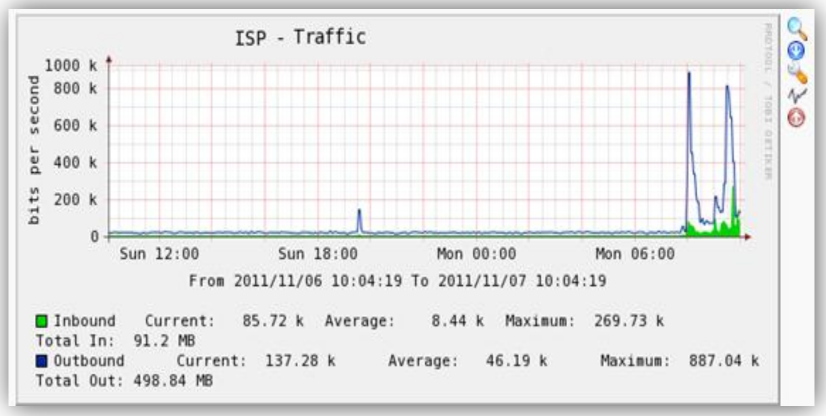


Monitor your network
…detailed graphs

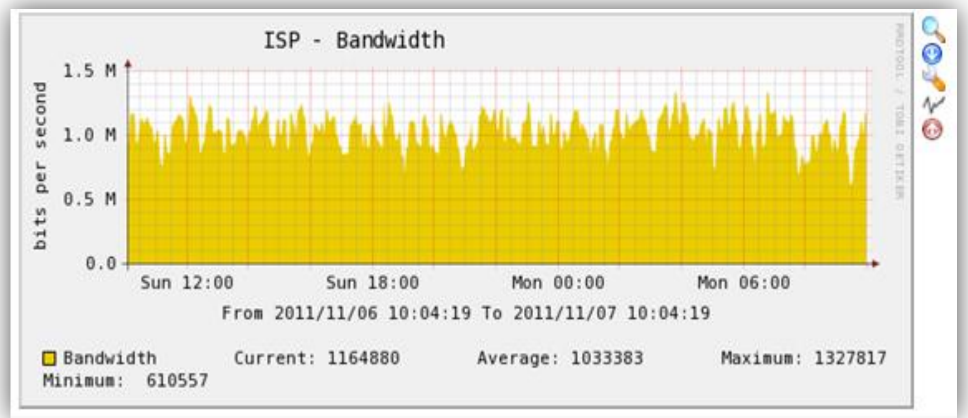
Latency



Inbound – Outbound usage



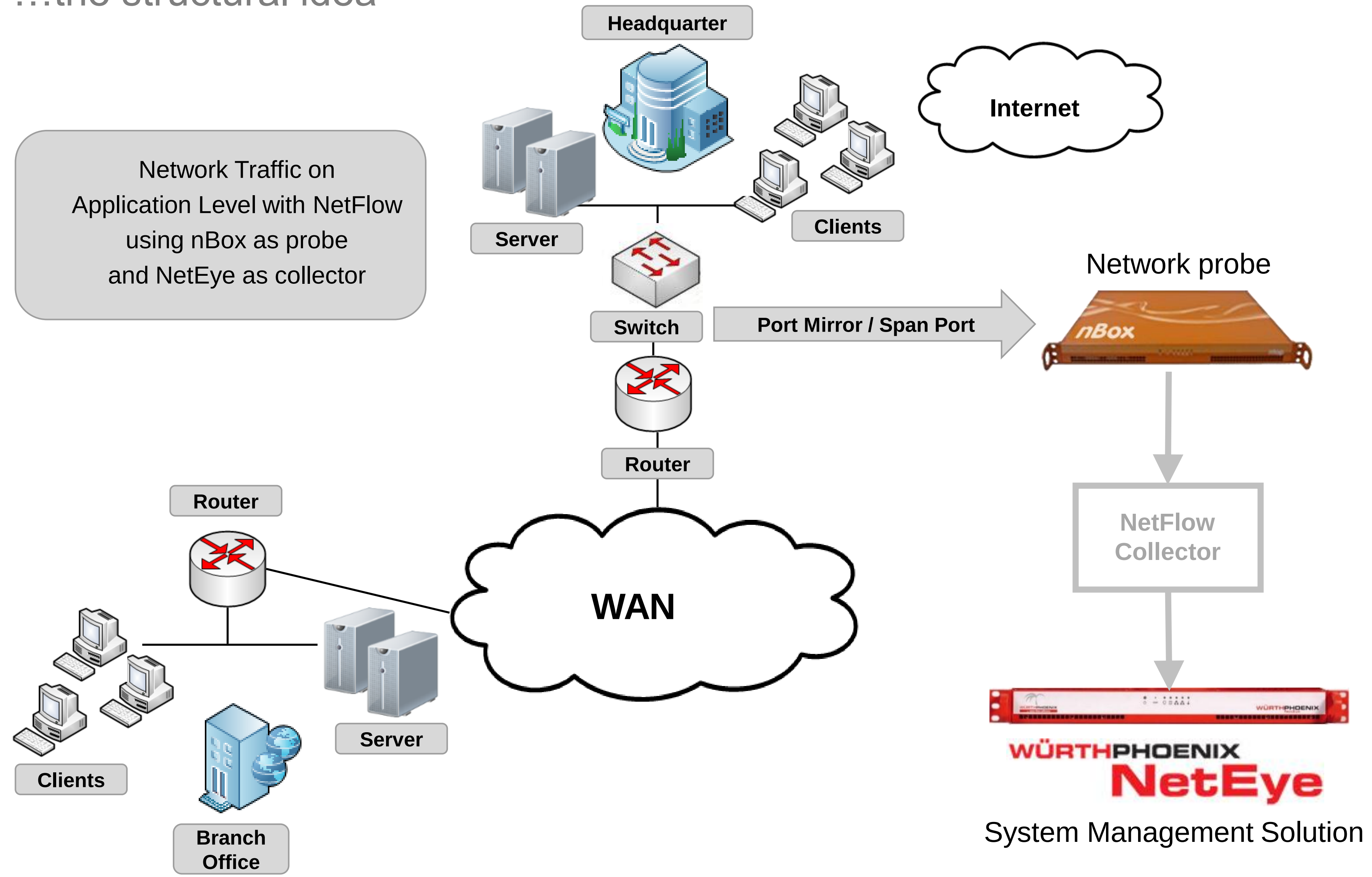
Bandwidth



Network Traffic Monitoring with NetFlow

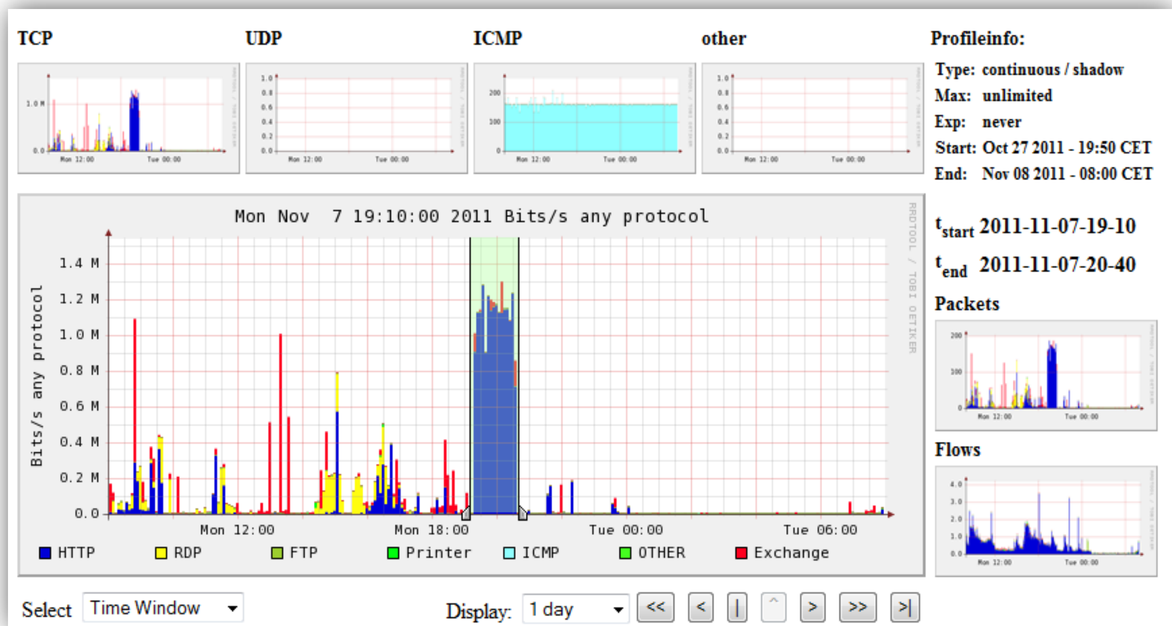
…the structural idea

Network Traffic on
Application Level with NetFlow
using nBox as probe
and NetEye as collector



Network Traffic Monitoring

…details on packets, bytes and ip/port



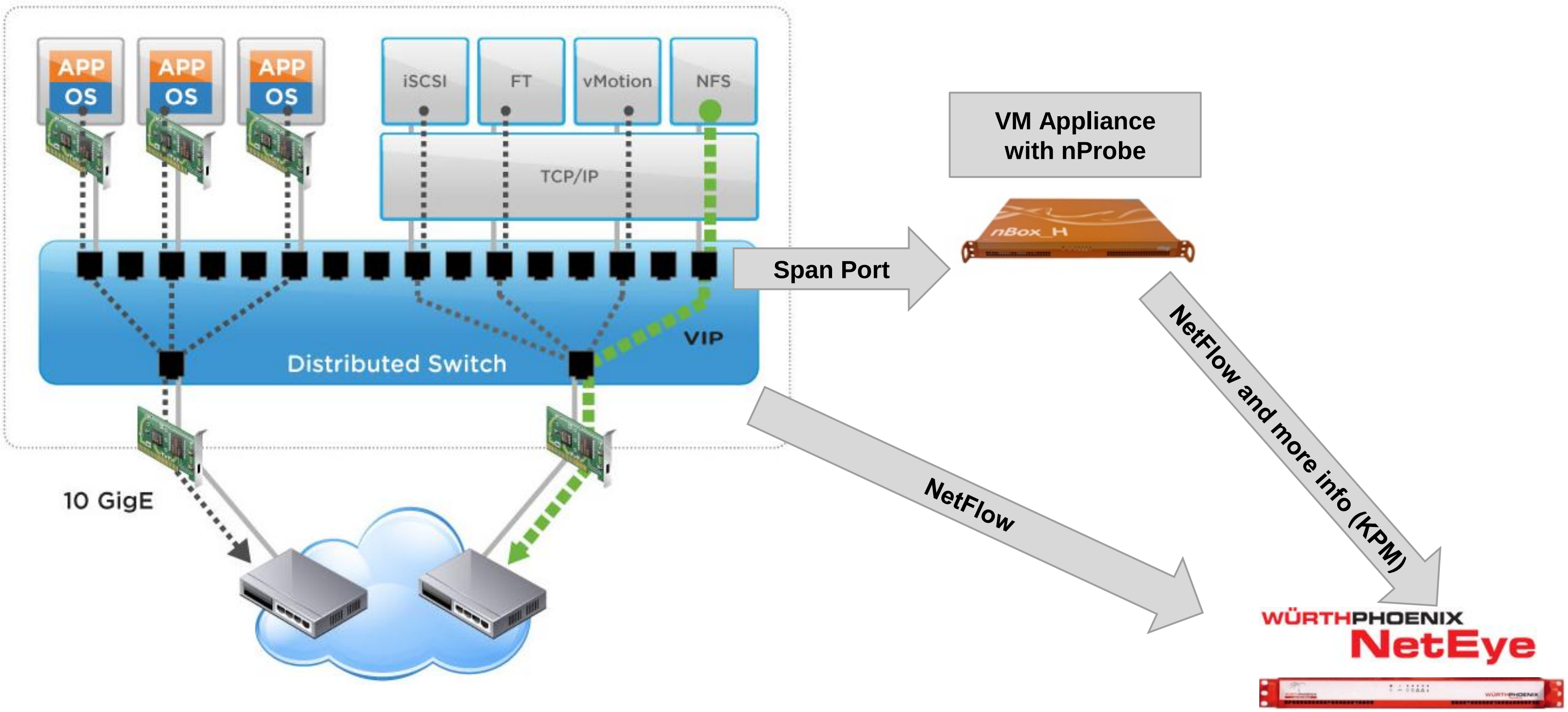
- Network traffic analysis based on protocols
- Source IP and Destination IP identification
- Filtering on single TCP / UDP ports
- Capability of network analyzing on packets, bytes per ip/port

Top 10 flows ordered by bytes:

Date flow start	Duration	Proto	Src IP Addr:Port		Dst IP Addr:Port	Flags	Tos	Packets	Bytes	Flows
2011-11-07 19:19:52.856	4670.430	TCP	10.62.1.91:33964	->	10.67.10.2:443	.APRSF	0	444663	624.2 M	152
2011-11-07 19:19:53.063	4670.242	TCP	10.67.10.2:443	->	10.62.1.91:34330	.AP.SF	0	45513	19.1 M	152
2011-11-07 19:19:52.869	4670.418	TCP	10.67.10.2:443	->	10.62.1.91:33964	.AP.SF	0	222389	11.6 M	152
2011-11-07 20:12:38.499	30.188	TCP	10.62.1.66:49741	->	10.67.10.2:25	.AP.SF	0	4252	6.3 M	2
2011-11-07 20:34:49.174	23.697	TCP	10.62.1.66:50425	->	10.67.10.2:25	.AP.SF	0	3466	5.2 M	2
2011-11-07 19:19:53.972	13.393	TCP	10.62.1.66:48113	->	10.67.10.2:25	.AP.SF	0	2485	3.7 M	2
2011-11-07 19:19:53.042	4670.263	TCP	10.62.1.91:34330	->	10.67.10.2:443	.APRSF	0	44739	2.4 M	152
2011-11-07 19:52:53.356	8.910	TCP	10.62.1.66:49148	->	10.67.10.2:25	.AP.SF	0	1312	1.9 M	2
2011-11-07 19:58:37.980	4.359	TCP	10.62.1.66:49323	->	10.67.10.2:25	.AP.SF	0	626	893300	2
2011-11-07 19:53:49.626	1439.270	TCP	10.62.1.91:58125	->	10.67.10.2:443	.AP.S.	0	966	620088	36

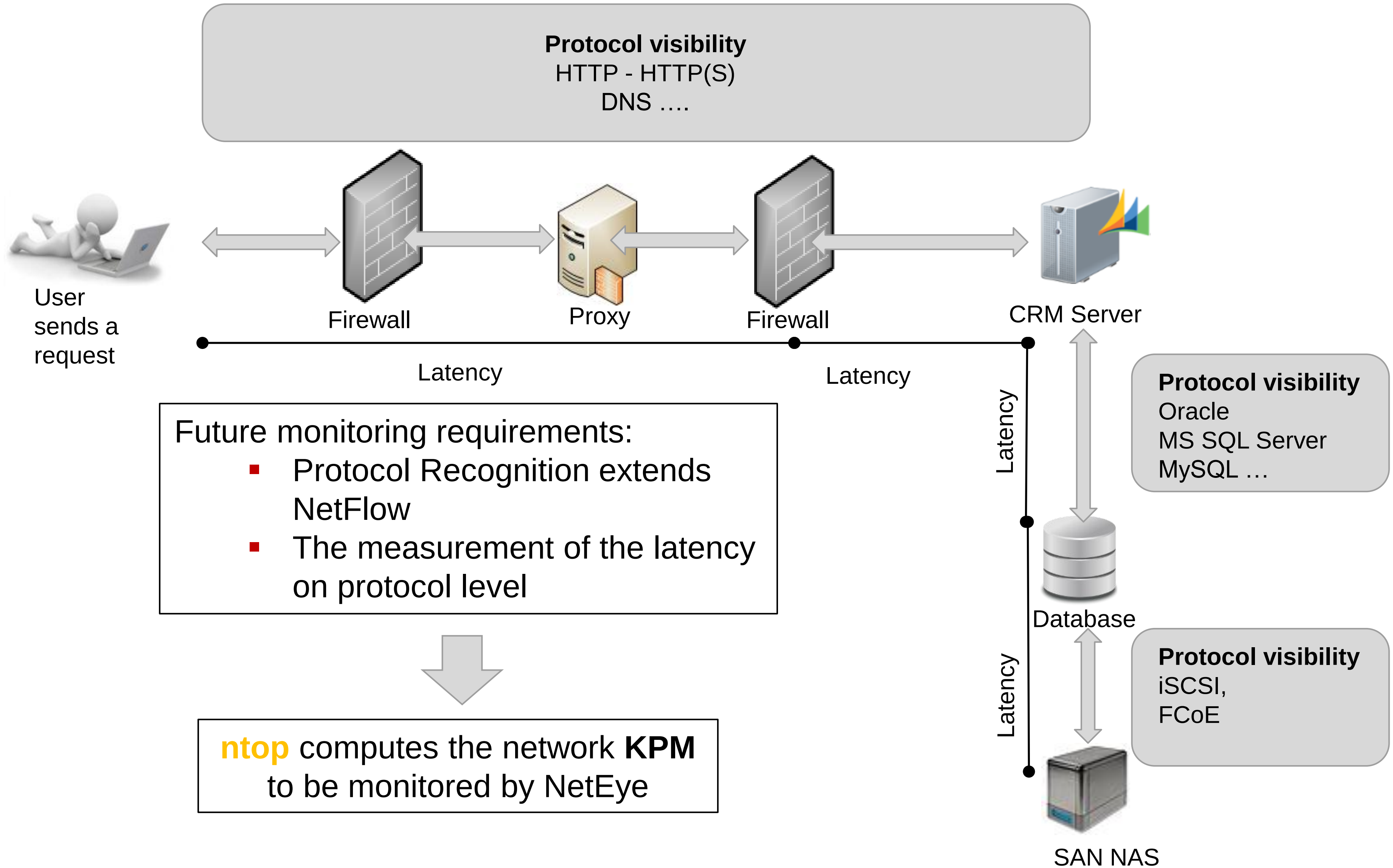
Network visibility into virtualized infrastructure

…collection of NetFlow and Key Performance Measures



Future monitoring targets…

What is causing slow performance? Network or Application?



Real User Monitoring

…the approach of WÜRTHPHOENIX NetEye

NetEye provides Real User Monitoring thanks to KPM metrics from nProbe:

Application Latency Monitoring measures the response time of each user transaction analyzing the communication performance to get three key performance indicators

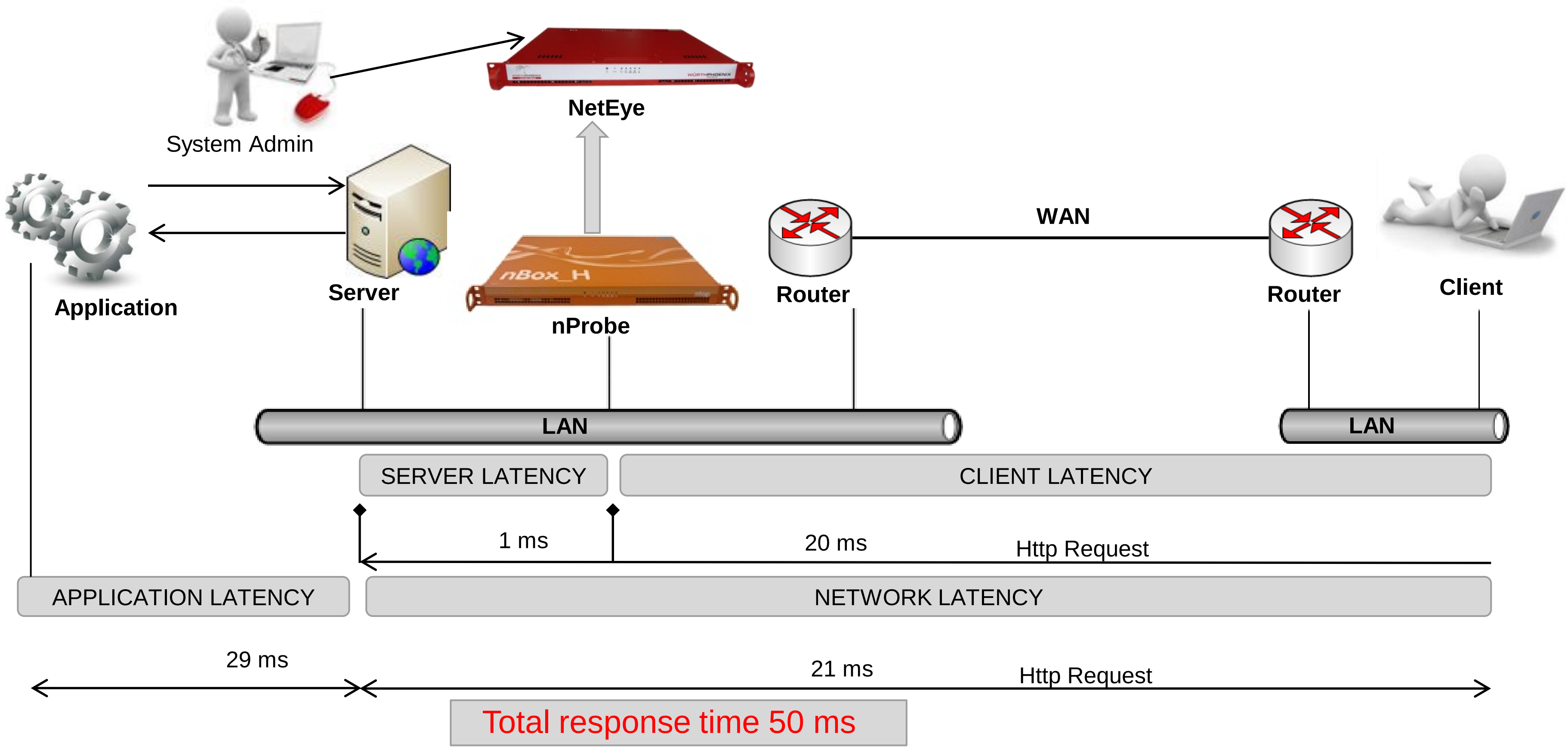
- Client Network Latency
- Server Network Latency
- Application Latency

Users' experience

End User Latency Monitoring

…how the response time is calculated

- Cycling monitoring is computed on **Client Network, Server Network, Application Latency** for each **End User** requests to discover slowness on network latency or applications.
- System alerts are generated on deviation from the normal End User performance



Alerts generated on latency deviation

…how to record the baselines

- The system runs for couple of days in normal network and application conditions to record the **baselines**
- The system calculates the average client/server/application latency based on the requests in the defined period
- At this point a periodic check runs (i.e. every 5 minutes) comparing the average latency with those of the relative baselines
- **Warning** and **critical** are generated based on customizable **thresholds percentage**
- **Minimum** and **maximum watermarks** can also be configured to create reasonable statistics (i.e. if the average latency are very low values (5ms), the percentage are not a reliable mechanism for the check)

Baselines definition

- Calculation of the average Client/ Server/Application latency
- Warning and critical notifications based on thresholds percentage

Application Latency Monitoring

…recorded baselines

File Edit Window

Application Latency Monitor

Baseline | Query | Graphs | Status | Configuration

Baseline

	Id	Name	Description	App Lat	Server Lat	Client Lat	Bytes	Sessions/mi	Sessions	Attempt	Flapping	Last Check	Status
...	Nessun filtro applicato												
☐	7		Scar App	0.154	3.558	96.988	13136.000	92.000	462	0	3	02/11/11 09:50:01	CRITICAL
☐	2		Dropbox	55.675	0.000	0.000	854.000	1.000	8	0	0	02/11/11 09:50:01	OK
☐	3		Facebook	0.249	26.812	0.090	10957.000	9.000	49	0	0	02/11/11 09:50:01	OK
☐	5		Main Web	0.061	0.151	18.070	25843.000	33.000	165	0	0	02/11/11 09:50:01	OK
☐	6		Trendmicro Update	0.213	5.756	1.330	4480.000	24.000	121	0	0	02/11/11 09:50:01	OK
☐	8		Fime App	0.022	0.296	0.276	8049.000	100.000	501	0	0	02/11/11 09:50:01	OK
☐	13		NetEye Updates	0.001	0.239	20.817	1931.000	14.000	73	0	0	02/11/11 09:50:01	OK
☐	14		CIS	0.128	41.542	0.173	4148.000	62.000	313	0	4	02/11/11 09:50:01	OK
☐	16		NetEye Blog	0.187	0.090	32.738	24184.000	1.000	8	0	0	02/11/11 09:50:01	OK
☐	17		Sylvestrix	0.100	0.433	1.517	2340.000	30.000	153	0	0	02/11/11 09:50:01	OK

1 - 35 di 35 elementi

Filter: *Filter shown baselines*

Refresh Baseline | Check Baseline | Show Rrd for selected

Monitoring metrics for each Application

Baseline: 0.095 **Actual:** -65% **Average of:** 5m
Min: 30.000 **Max:** 80.000 **Ref:** 0.095
Warning: + 10% (0.105) **Critical:** + 20% (0.115)
Value under minimum allowed => OK

Latency indicators

…aggregated by locations

Netgroup/Application/Subnet/Client	Requests	App Latency	Server Latency	Client Latency	Bytes	Status
⊞ Bolzano	8862	0.437	21.762	4.100	67,2M	
⊟ Roma	3640	0.510	0.255	18.153	97,3M	
⊞ UNMATCHED	3308	0.509	0.252	18.762	92,6M	
⊟ Facebook	124	1.122	0.341	14.775	2,7M	
⊞ Skype	18	0.029	0.150	10.013	63,5k	
⊞ NetEye Updates	3	0.000	0.245	8.853	5,5k	
⊞ Google	187	0.169	0.267	10.548	1,8M	
⊟ VLAN 1 LAN	124	1.122	0.341	14.775	2,7M	
▪ 10.62.11.153	18	2.871	0.212	22.531	1,3M	
▪ 10.62.11.20	23	3.413	0.595	10.391	1,2M	
▪ 10.62.11.75	83	0.107	0.298	14.308	174,9k	

↓ Application
LAN
User IP

…aggregated by clients

Client/Subnet/Netgroup/Application	Requests	App Latency	Server Latency	Client Latency	Bytes	Status
⊞ 10.62.11.157	736	0.384	0.154	36.973	27,2M	CRITIC
⊞ 10.62.11.21	1	0.001	0.568	75.453	537,9b	CRITIC
⊞ 10.62.37.166	12	0.033	3.362	156.045	182,0k	CRITIC
⊞ 10.62.37.172	44	0.078	3.886	41.718	466,1k	CRITIC
⊞ 10.62.37.175	50	0.056	14.952	255.339	379,9k	CRITIC
⊞ 10.62.37.25	54	0.031	4.202	55.469	303,3k	CRITIC
⊞ 10.62.37.53	30	0.055	8.162	44.943	389,5k	CRITIC
⊞ 10.62.38.50	50	0.035	4.586	273.501	407,9k	CRITIC
⊞ 10.62.4.23	80	0.239	104.869	0.213	127,5k	CRITIC
⊞ 10.62.4.30	98	0.099	108.672	0.199	148,2k	CRITIC

Latency indicators

…aggregated by applications

	Netgroup/Application/Subnet/Client	Requests	App Latency	Server Latency	Bytes	Status
☐	⊞ Dropbox	10	55.710	88.424	8.4k	CRITICAL
☐	⊞ Facebook	143	10.639	57.389	1.0M	CRITICAL
☐	⊞ CIS	178	0.657	51.700	1.2M	CRITICAL
☐	⊞ Scar App	222	0.030	4.380	2.5M	WARNING
☐	⊞ UNMATCHED	3790	0.276	20.065	90.1M	OK
☐	⊞ Repubblica	21	0.023	5.162	183.6k	OK
☐	⊞ Main Web	7	0.172	0.102	65.0k	OK
☐	⊞ Trendmicro Update	37	0.050	4.882	1.5M	OK
☐	⊞ Fime App	75	0.005	0.321	128.1k	OK
☐	⊞ Skype	1	0.041	18.320	1.0k	OK
☐	⊞ NetEye Updates	43	0.000	0.422	83.2k	OK

Drill down to URL details

URL	From	Requests	App Latency	Server Latency	Client Latency	Bytes
http://googleads.g.doubleclick.net/p	Scar	1	0.690	11.582	10.987	3.2k
http://go.microsoft.com/fwlink/\%3F	Scar	2	0.182	88.396	5.713	3.1k
http://go.microsoft.com/fwlink/\%3F	Scar	7	0.181	88.201	4.943	12.4k
http://g.microsoft.com/_0sfdata/1\%	Scar	1	0.110	55.224	3.700	1.4k
http://fxfeeds.mozilla.com/it/firefox/	Scar	3	0.023	10.441	6.512	4.8k
http://emea.rel.msn.com/default.asp	Scar	[illegible]	[illegible]	[illegible]	4.020	1.4k
http://du106w.dub106.mail.live.com/	Scar	[illegible]	[illegible]	[illegible]	5.763	953.5k

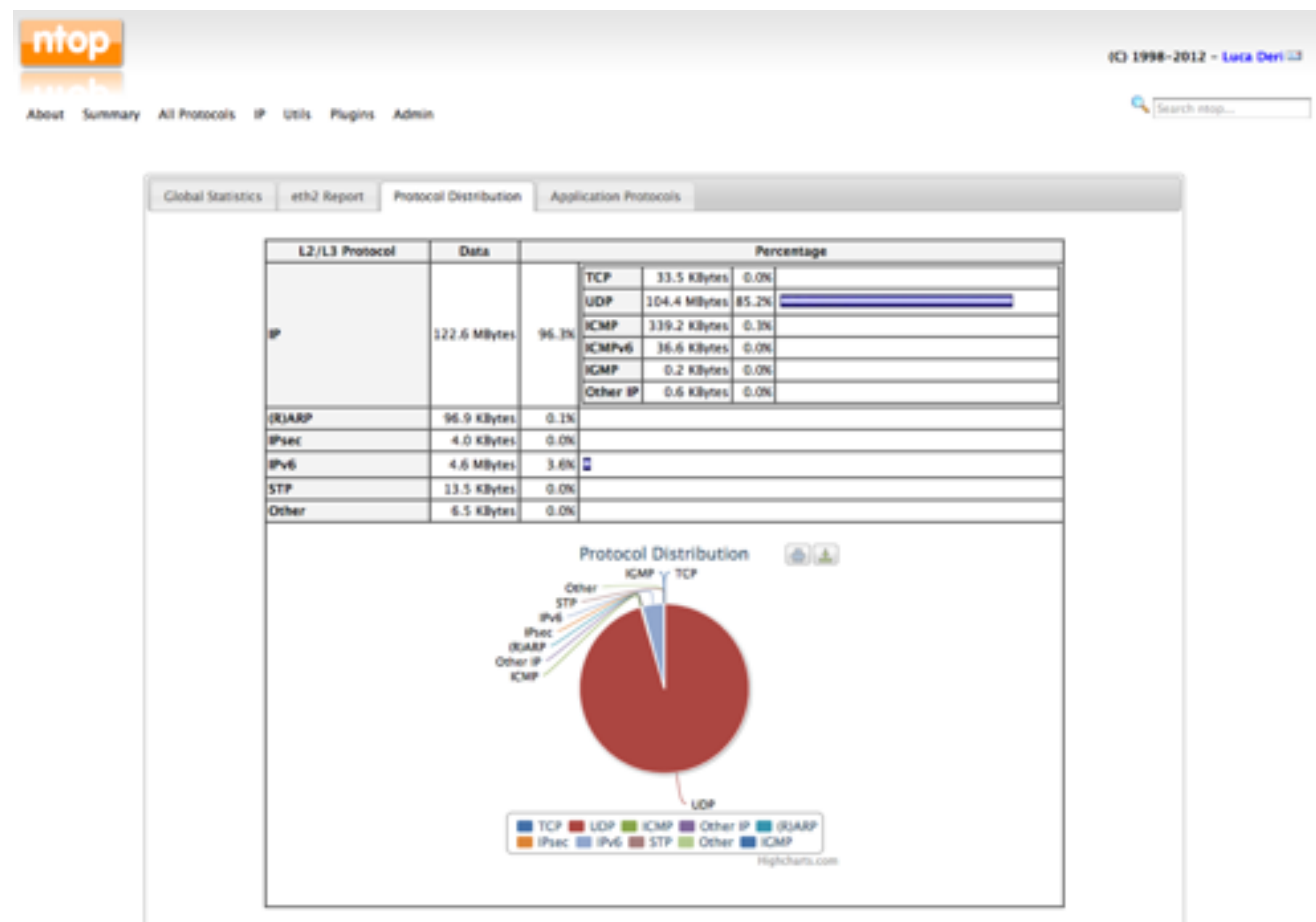
http://du106w.dub106.mail.live.com/

Mehr Klarheit für Ihre Netzwerk Performance

Luca Deri <deri@ntop.org>

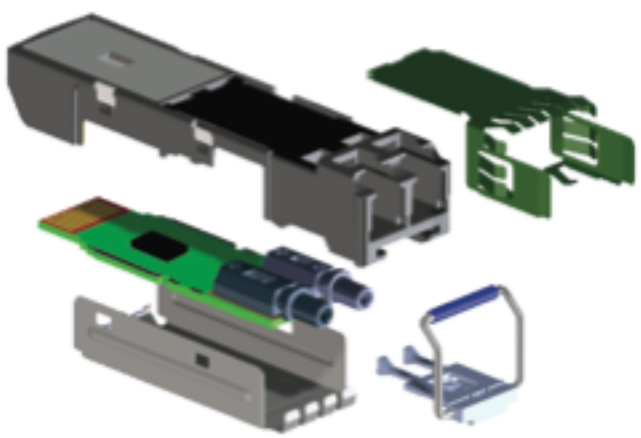
About ntop.org [1/3]

- Private company devoted to development of Open Source network traffic monitoring applications.
- ntop (circa 1998) is the first app we released and it is a web-based network monitoring application.



About ntop.org [2/3]

- Our software is powering many commercial products...



Integrated ASIC with JDSU technology

About ntop.org [3/3]

- ...and allows packets to be received and transmitted at 1/10 Gbit line rate with no loss, any packet size on commodity NICs developed by our partner
- So we accelerate not just our applications but also third party open source solutions including:

Problem Statement [2/3]

- Popular hardware probes (Juniper, Cisco) are costly, limited (usually no analysis beyond packet header) and often not extensible.
- Consequences:
  - Monitoring evolution is capped by hardware vendors.
  - Commercial probes monitor what the vendor <u>wants</u> (e.g. Cisco TelePresence) and not what the user <u>needs</u>.
  - The Internet is changing so fast (Twitter, YouTube, NetFlix...) that collectors relying on hardware probes cannot provide timely answers to continuously evolving monitoring needs.

ntop and Würth-Phoenix Goals

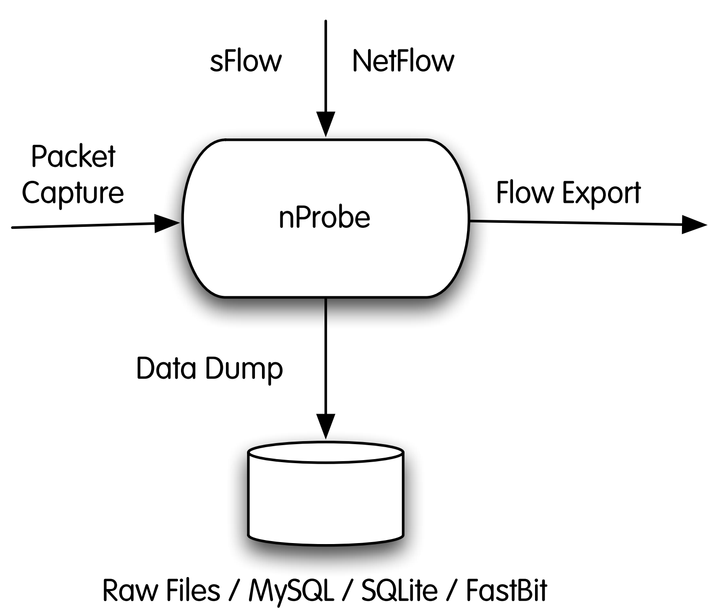
- Provide better, yet price effective, traffic monitoring by allowing collectors to have increased traffic visibility.
- NetEye will integrate thew new network metrics to report users about the probe findings.
- ntop+NetEye joint forces are the only way for producing comprehensive and accurate traffic reports able to offer at a fraction of price what many commercial products do together.

Monitoring Architecture

Traffic Flows

ntop nBox

Würth-Phoenix NetEye



Limitations of Monitoring Systems

- Visibility limited to packet header (payload agnostic).
- Packet encapsulations (e.g. GRE, PPP, GTP) are not always handled, so that we don't know what happens inside tunnels.
- Unable to monitor intra-virtual machines (VMs) traffic (no cloud-friendly).
- Windows PCs are not first-class citizens.

Why We're Different?

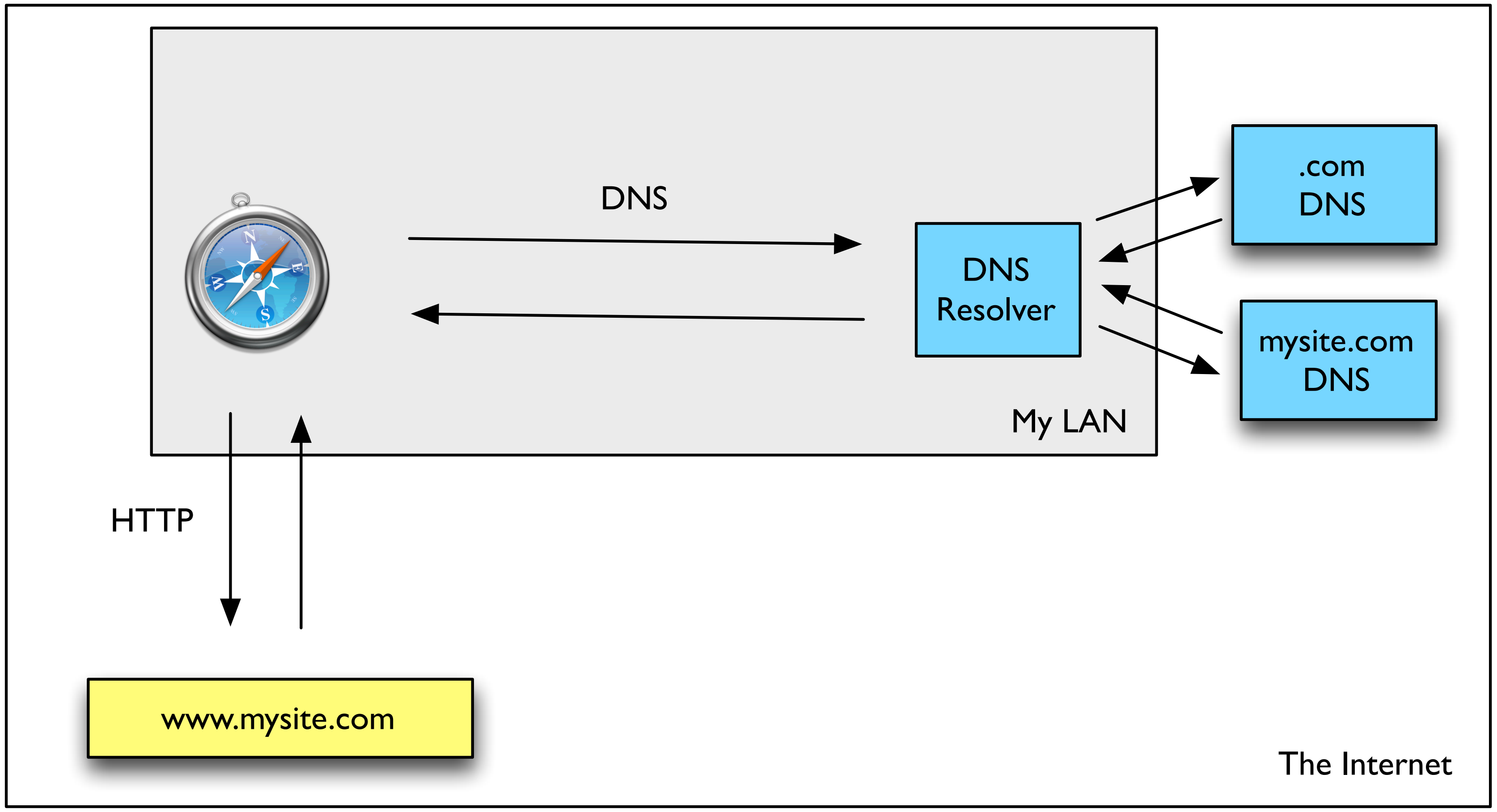
- We provide evidence of networks issues
  - You know exactly what has happened.
- We measure KPM (Key Performance Metrics)
  - You know the health of your network services.
- We recognize network protocols
  - We tell you exactly what applications are using your network.
- We compute your network trends
  - We provide you a forecast for growing your network before it's too late.

Providing Evidence [1/2]

- Network administrators often receive generic issue reports:
  - *"Internet browsing is slow and often URLs cannot be accessed. Trying again usually helps for visiting the target web site."*
- Flow-based traffic analysis provides an <u>average</u> view of a network communication.
- Network services (e.g. web surfing) are in good state when all components are healthy.

Providing Evidence [2/2]

- Simple actions such as web surfing require the interaction of various actors.
- One a few of them are under our control.



Providing Evidence: DNS

[NFv9 57677][IPFIX 35632.205] %DNS_QUERY	DNS query
[NFv9 57678][IPFIX 35632.206] %DNS_QUERY_ID	DNS query transaction Id
[NFv9 57679][IPFIX 35632.207] %DNS_QUERY_TYPE	DNS query type (e.g. 1=A, 2=NS..)
[NFv9 57680][IPFIX 35632.208] %DNS_RET_CODE	DNS return code (e.g. 0=no error)
[NFv9 57681][IPFIX 35632.209] %DNS_NUM_ANSWER	DNS # of returned answers
[NFv9 57558][IPFIX 35632.86] %APPL_LATENCY_SEC	Application latency (sec)
[NFv9 57559][IPFIX 35632.87] %APPL_LATENCY_USEC	Application latency (usec)

```
#
# When|DNS_Client|AS|ClientCountry|ClientCity|DNS_Server|Query|NumRetCode|RetCode|
NumAnswer|NumQueryType|QueryType|TransactionId|Answers|AuthNSs
#
1326819546.137|A.B.C.D|XXXX|US||192.12.192.5|blogsearch.google.it|0|NOERROR|0|1|A|52017||
ns2.google.com;ns1.google.com;ns4.google.com;ns3.google.com
```

Providing Evidence: HTTP

[NFv9 57652][IPFIX 35632.180] %HTTP_URL	HTTP URL
[NFv9 57653][IPFIX 35632.181] %HTTP_RET_CODE	HTTP return code (e.g. 200, 304...)
[NFv9 57654][IPFIX 35632.182] %HTTP_REFERER	HTTP Referer
[NFv9 57655][IPFIX 35632.183] %HTTP_UA	HTTP User Agent
[NFv9 57656][IPFIX 35632.184] %HTTP_MIME	HTTP Mime Type

```
#
# Client    Server  Protocol    Method  URL    HTTPReturnCode  Location    Referer UserAgent
ContentType    Bytes   BeginTime    EndTime Flow Hash    Cookie  Terminator    ApplLatency(ms)
ClientLatency(ms)    ServerLatency(ms)    Application
#
192.168.0.200   www.macintouch.com    http    GET    /images/filewave01.gif  200
www.macintouch.com    Mozilla/5.0 (Macintosh; U; PPC Mac OS X; en) AppleWebKit/416.12 (KHTML,
like Gecko) Safari/416.13
    27750   1133966828.928  1133966830.606  26992029    0    S    0.261   0.080   114.095 HTTP
```

Providing Evidence: VoIP

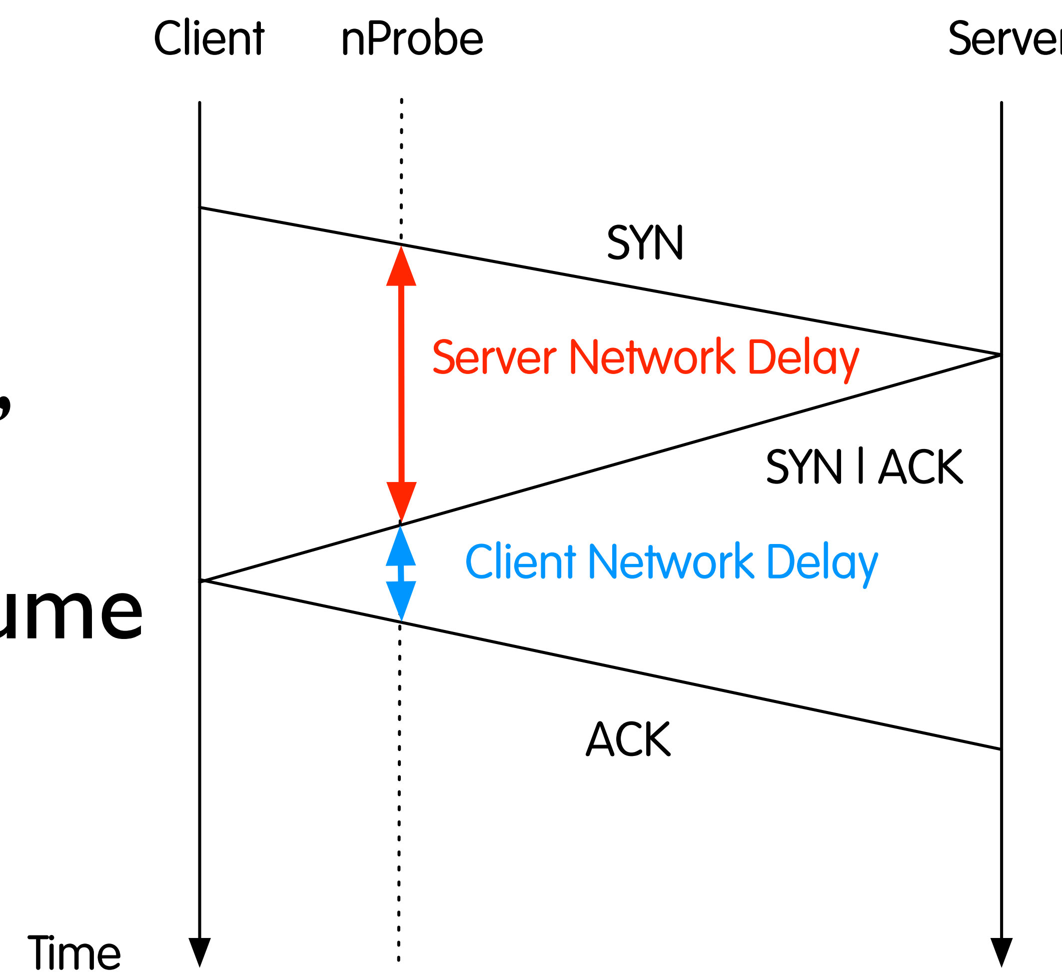
[NFv9 57602][IPFIX 35632.130] %SIP_CALL_ID	SIP call-id
[NFv9 57603][IPFIX 35632.131] %SIP_CALLING_PARTY	SIP Call initiator
[NFv9 57604][IPFIX 35632.132] %SIP_CALLED_PARTY	SIP Called party
[NFv9 57605][IPFIX 35632.133] %SIP_RTP_CODECS	SIP RTP codecs
[NFv9 57606][IPFIX 35632.134] %SIP_INVITE_TIME	SIP SysUptime (msec) of INVITE
[NFv9 57607][IPFIX 35632.135] %SIP_TRYING_TIME	SIP SysUptime (msec) of Trying
[NFv9 57608][IPFIX 35632.136] %SIP_RINGING_TIME	SIP SysUptime (msec) of RINGING
[NFv9 57609][IPFIX 35632.137] %SIP_INVITE_OK_TIME	SIP SysUptime (msec) of INVITE OK
[NFv9 57610][IPFIX 35632.138] %SIP_INVITE_FAILURE_TIME	SIP SysUptime (msec) of INVITE FAILURE
[NFv9 57611][IPFIX 35632.139] %SIP_BYE_TIME	SIP SysUptime (msec) of BYE
[NFv9 57612][IPFIX 35632.140] %SIP_BYE_OK_TIME	SIP SysUptime (msec) of BYE OK
[NFv9 57613][IPFIX 35632.141] %SIP_CANCEL_TIME	SIP SysUptime (msec) of CANCEL
[NFv9 57614][IPFIX 35632.142] %SIP_CANCEL_OK_TIME	SIP SysUptime (msec) of CANCEL OK
[NFv9 57615][IPFIX 35632.143] %SIP_RTP_IPV4_SRC_ADDR	SIP RTP stream source IP
[NFv9 57616][IPFIX 35632.144] %SIP_RTP_L4_SRC_PORT	SIP RTP stream source port
[NFv9 57617][IPFIX 35632.145] %SIP_RTP_IPV4_DST_ADDR	SIP RTP stream dest IP
[NFv9 57618][IPFIX 35632.146] %SIP_RTP_L4_DST_PORT	SIP RTP stream dest port
[NFv9 57619][IPFIX 35632.147] %SIP_FAILURE_CODE	SIP failure response code
[NFv9 57620][IPFIX 35632.148] %SIP_REASON_CAUSE	SIP Cancel/Bye/Failure reason cause

Providing Evidence: MySQL

[NFv9 57667][IPFIX 35632.195] %MYSQL_SERVER_VERSION	MySQL server version
[NFv9 57668][IPFIX 35632.196] %MYSQL_USERNAME	MySQL username
[NFv9 57669][IPFIX 35632.197] %MYSQL_DB	MySQL database in use
[NFv9 57670][IPFIX 35632.198] %MYSQL_QUERY	MySQL Query
[NFv9 57671][IPFIX 35632.199] %MYSQL_RESPONSE	MySQL server response

KPM: Network and Application Performance [1/3]

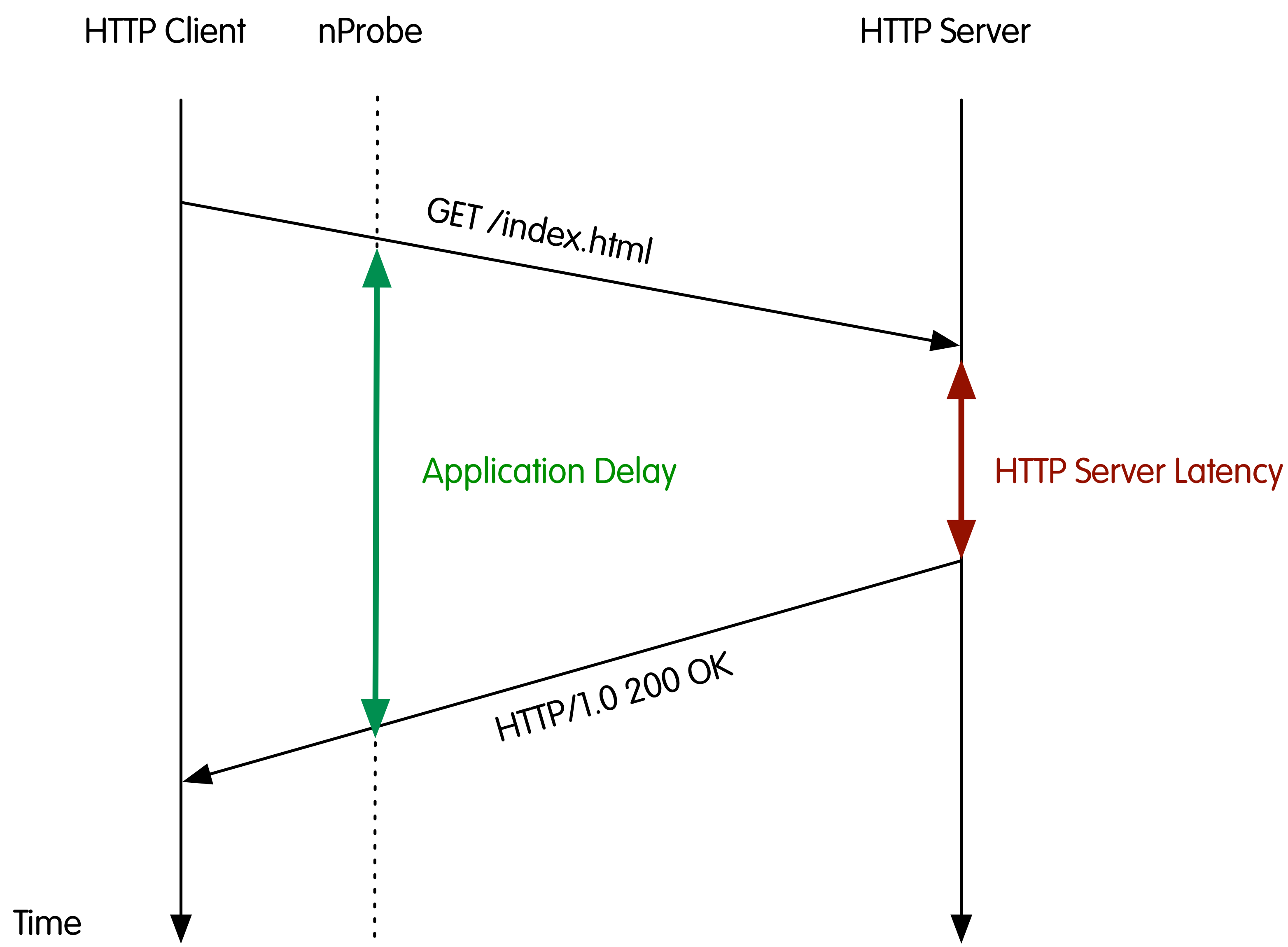
- Client and server network delay are determined when the nProbe observes the TCP flags in a transaction.
- Simple 3 packet transaction (TCP only).
- Divide the time delta by two, as we want to compute the network latency that we assume is half the round trip time.



KPM: Network and Application Performance [2/3]

- Application latency is computed as the time needed by an application to react to a client request.
- For TCP connections, application latency is computed on the first packet after three-way-handshake.
- For UDP connections on the first client-to-server and server-to-client packet.

KPM: Network and Application Performance [3/3]



HTTP Server Latency = Application Delay - Server Network Delay

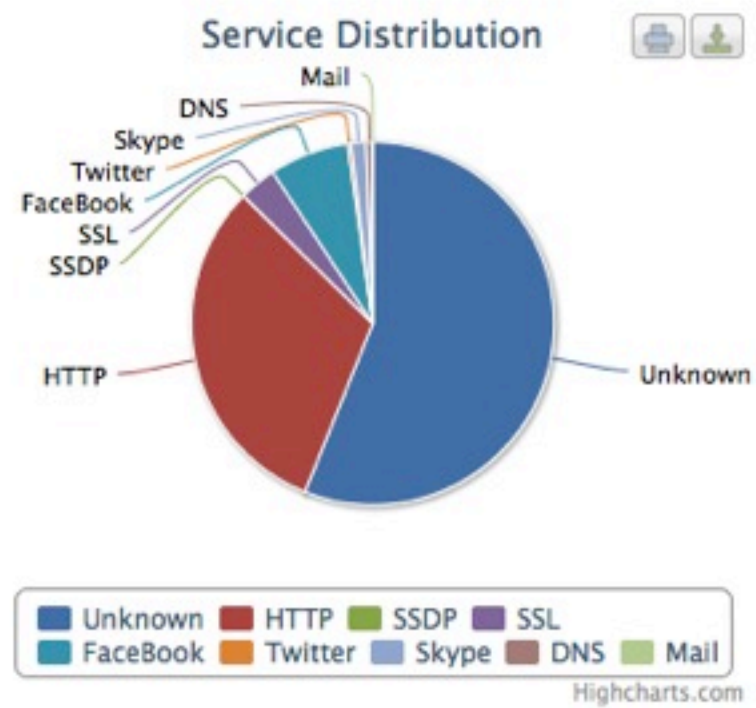
Protocol Recognition [1/3]

- Recognizing protocols is necessary for many reasons:
  - Know what protocols are occupying the network for good (business) or bad (leisure) reasons.
  - Double-check if claims done when deploying services are really true (e.g. protocol X uses little bandwidth)
  - Identify security flaws (e.g. long-standing SSL/SSH connections).
  - Detect violation of network policies (e.g. well known protocols on non-standard ports).

Protocol Recognition [2/3]

- nDPI is a DPI library based on OpenDPI (GPL) to which:
  - We have added several new protocols (e.g. YouTube, Skype, Twitter, FaceBook, Citrix, SSL, email) for a total of over 130 protocols in total.
  - We have made some code changes that made it faster for our network monitoring needs.
  - It can use the protocol+port as fallback in case DPI is not applicable (e.g. we have missed the initial 3-way handshake).

Protocol Recognition [3/3]



Application Protocol	Data	Percentage	
Unknown	5.7 MBytes	56.1%	
HTTP	3.2 MBytes	31.4%	
SSDP	1.4 KBytes	0%	
SSL	322.2 KBytes	3.1%	
FaceBook	740.2 KBytes	7.1%	
Twitter	32.1 KBytes	0%	
Skype	132.3 KBytes	1.3%	
DNS	64.4 KBytes	0%	
Mail	0.7 KBytes	0%	

Combining Protocol Recognition with Performance

- KPMs are selected based on the protocol.
  - Low network latency (i.e. network delay) is required by interactive (e.g. SSH) and multimedia (e.g. VoIP) protocols.
  - High throughput is desirable for data transfer protocols (e.g. file transfer).
- NetEye can be used to produce alerts based on the protocol so that alarms are emitted only when it make sense.

Network Trends [1/2]

- For years monitoring systems have computed network trends only based on packets and bytes.
- While this practice is correct, it limits its scope to network bandwidth growth.
- As we measure many KPMs, we can finally put an eye also on many other indicators.
- This allows network administrators to also evaluate how changes they do on the network affect user's network experience.

Network Trends [2/2]

- A list of trends we measure include (but are not limited to):
  - Network latency.
  - Application response time.
  - Packet loss.
  - Jitter (VoIP).
  - TTL (number of hops to a destination).
  - Packet retransmissions.
  - Packets out-of-order.

Evaluating Traffic Quality

- We're developing a numeric nProbe flow quality index that represents various flow aspects:
  - Packet quality (e.g. good 3-way handshake, fragments).
  - Flow bandwidth (linear or fuzzy flow throughput).
  - Known protocols on non standard ports (are people trying to circumvent network policies?).
  - Traffic exchanged with malware sites (e.g. integration with blacklists from malwaredomains.com).
  - Flow health (e.g. HTTP flows that have been stopped, or with unexpected latency).

Summary

- Accurate protocol and performance measurement is the key difference between ntop +NetEye and similar solutions.
- Precise problem report and rich KPMs production give network administrator a comprehensive view of their network.
- Capitalizing on open-source grants quick product evolution and ability to add extensions, otherwise not possible with closed-source products.