RIPE NCC
Dubai

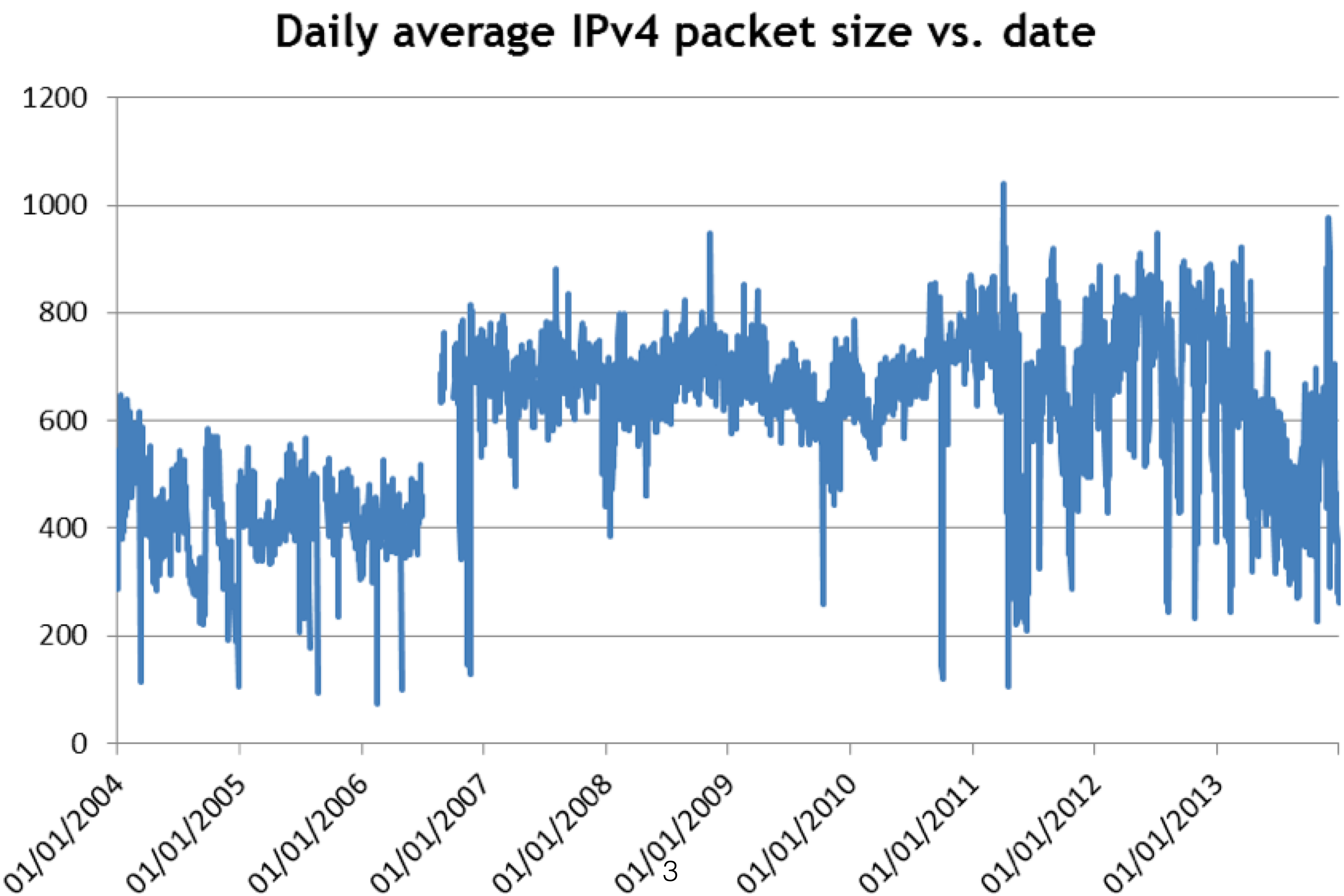
22-26 October 2017

Introducing an almost reliable UDP protocol: The Keyed UDP

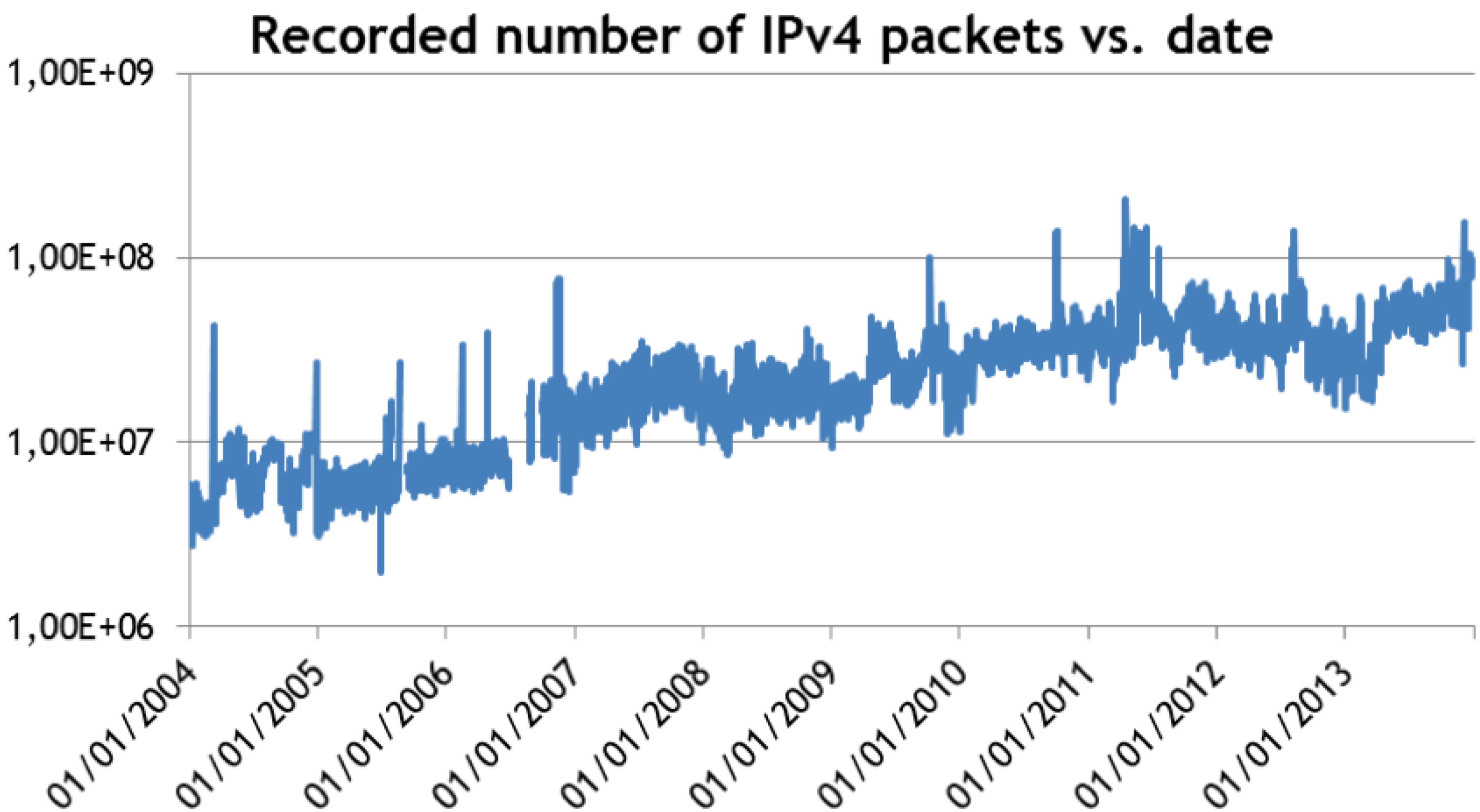
Nuno M. Garcia, *PhD*
Coordinator of Assisted Living Computing and Telecommunications Laboratory
Chair of COST Action IC 1303

Does this looks familiar?

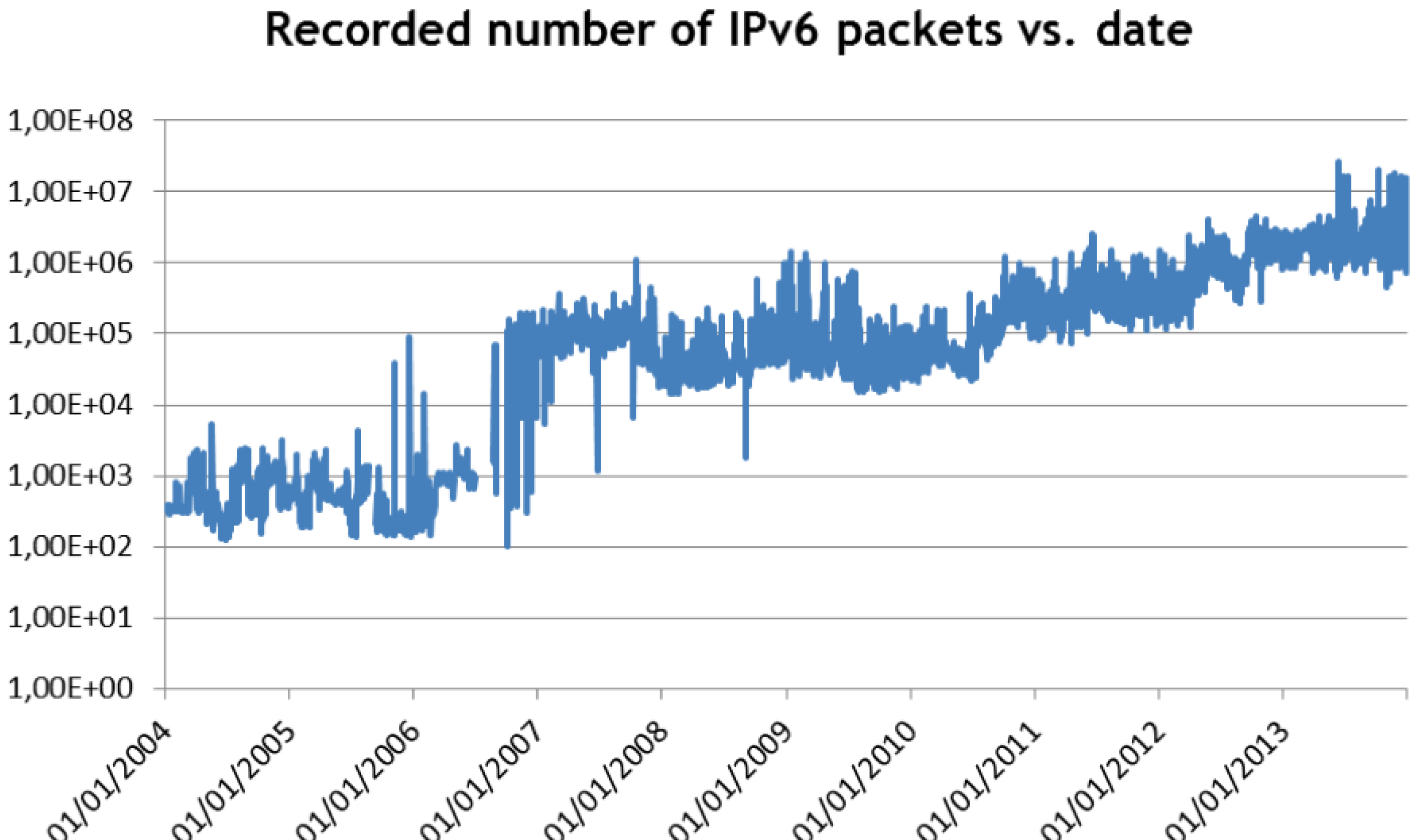
What about this?



Quick overview of the Internet traffic in the last 10 years

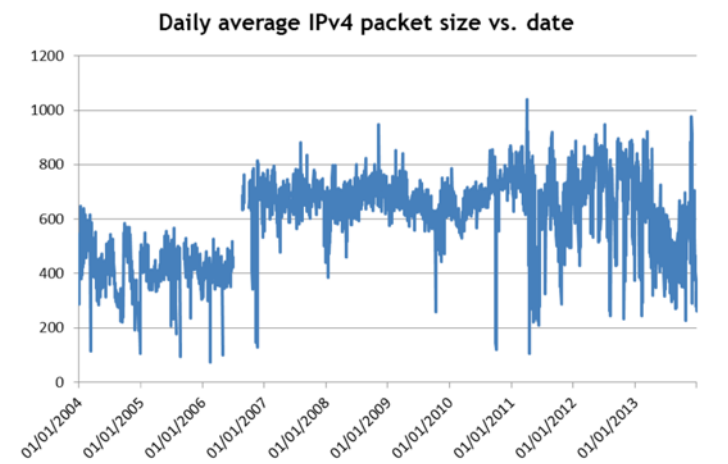


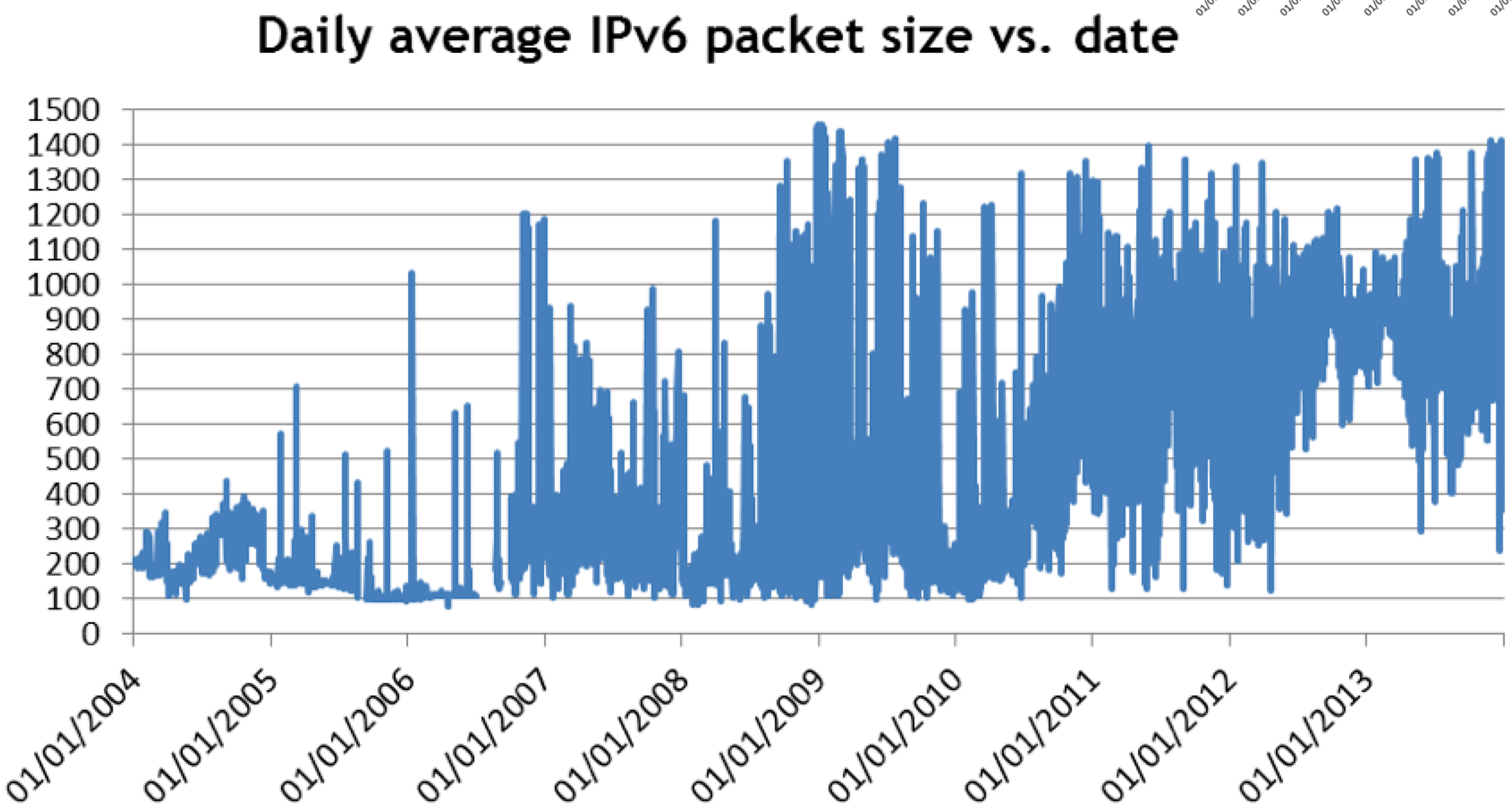
evolution++



Recorded number of IPv6 packets vs. date
1,00E+08
1,00E+07
1,00E+06
1,00E+05
1,00E+04
1,00E+03
1,00E+02
1,00E+01
1,00E+00
01/01/2004
01/01/2005
01/01/2006
01/01/2007
01/01/2008
01/01/2009
01/01/2010
01/01/2011
01/01/2012
01/01/2013

evolution++





Nuno M. Garcia – ngarcia@di.ubi.pt

And what about applications?

- We have witnessed:
  - more social media
  - more Cloud
  - more X as a Service
    - (virtualization of desktop on the User Side Resources to the Cloud - why?)

Nuno M. Garcia – ngarcia@di.ubi.pt

Applications in the future

- What can we expect to see in the future?
  - mobile phones now host octacores
  - mobile operators sell data packets
  - every device has some Web App or some Internet service connection (***e.g.*** problems with hacked cars)
  - my wristwatch monitors me, my car monitors me, my TV monitors me, my refrigerator monitors everyone, …
  - “every device will be a screen to The One” (Kevin Kelly, The first 5000 days of the Web)

Nuno M. Garcia – ngarcia@di.ubi.pt

Applications in the future

- User driven (social media, games, apps…)
- Government driven (eGovernment, eCitizenship, …)
- Corporations driven (eCustomer)
- Machine driven (P2P, machines that are Internet connected)
- Open source hardware (Arduíno, Raspberry Pi-Like, etc.)
- Open source software (Linux, Python, etc.)

Nuno M. Garcia – ngarcia@di.ubi.pt

What can we expect?

Nuno M. Garcia – ngarcia@di.ubi.pt

What can we do?

- *Problem 1: The Ethernet payload limit (99.999% of the measured packets that travel in the Internet are < 1500B)*
- *Problem 2: The TCP control mechanism (Three Way Handshake, AIMD)*
- *Problem 3: low efficiency of transport protocols (high overhead)*
- *Problem 4: increase in the size of messages*
- *Problem 5: transition to IPv6*
- *Problem 6: increased mobile traffic*

Nuno M. Garcia – ngarcia@di.ubi.pt

Really, what can we do?

- We need new protocols for Machine-to-Machine communication, for real time data transmission, which integrate data imputation, resiliency on the software side, not on the network side

Nuno M. Garcia – ngarcia@di.ubi.pt

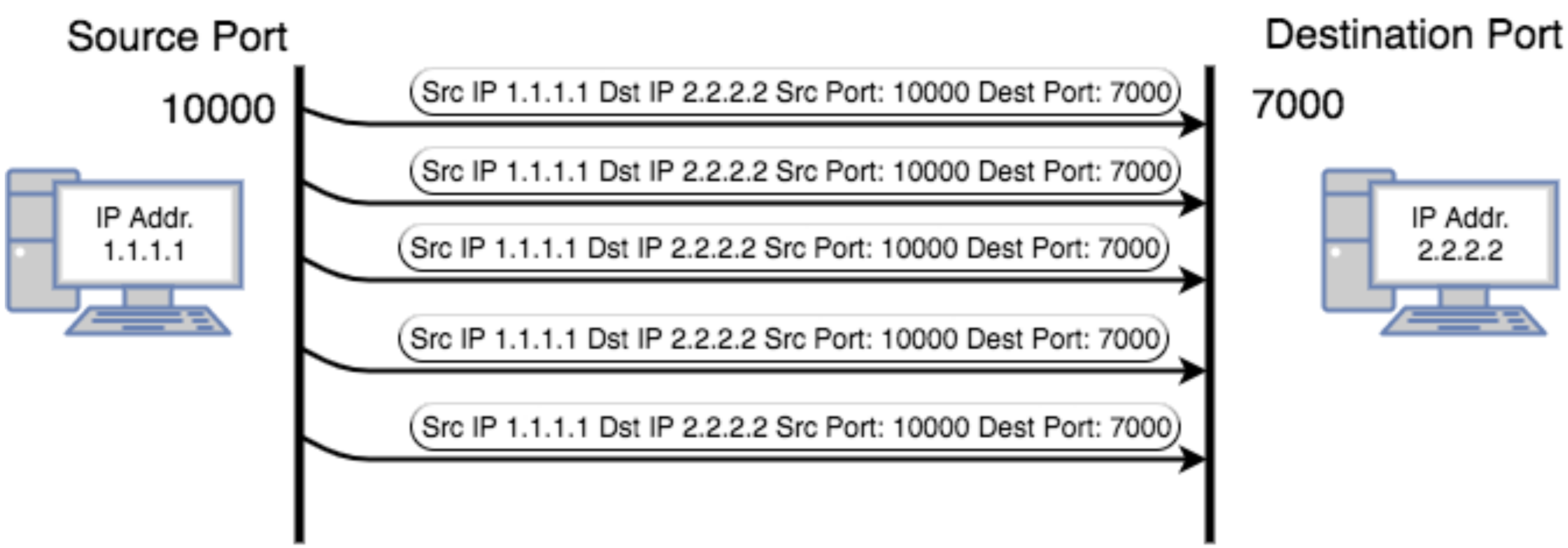
New Keyed-UDP

- UDP is faster and lighter than TCP
- Let us use UDP in a manner that sender and/or receiver can know, to a degree of certainty, if there was packet loss and what packets were lost
- I call this Keyed UDP

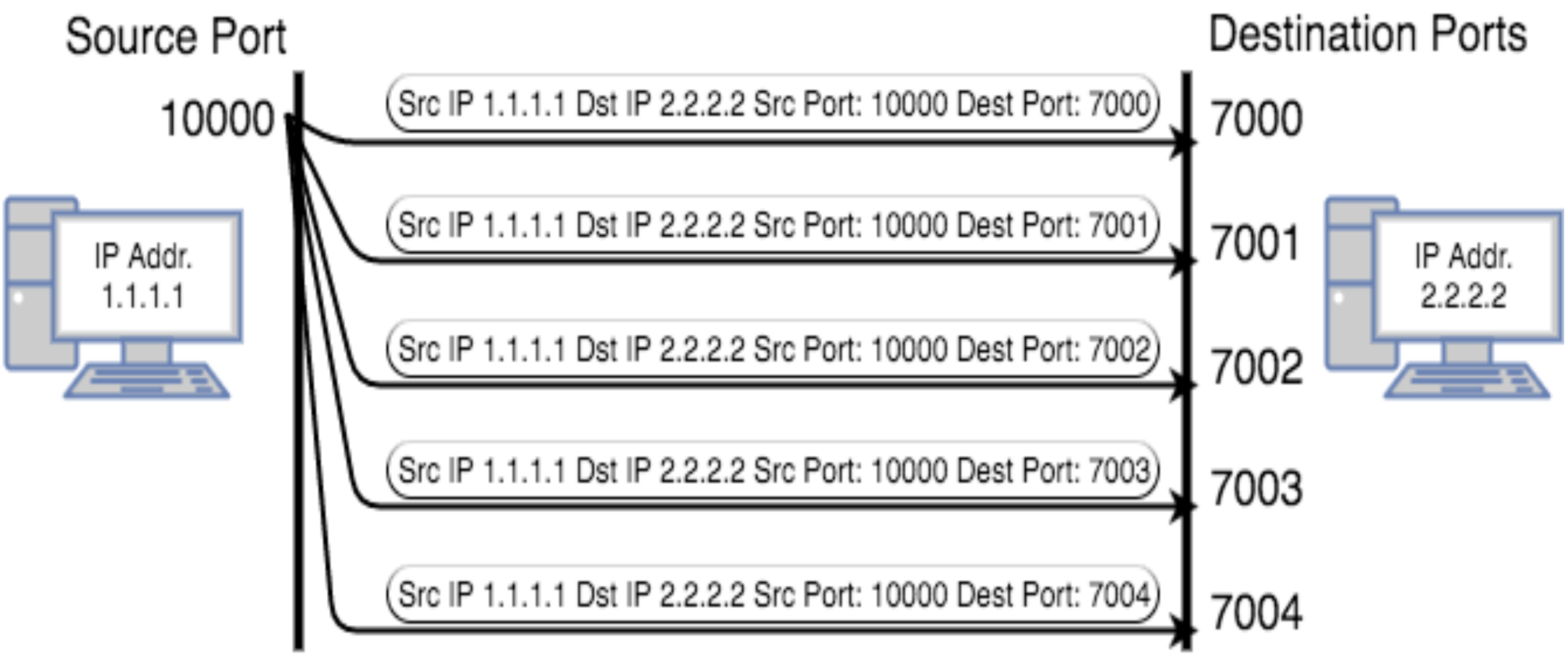
Nuno M. Garcia – ngarcia@di.ubi.pt

Keyed-UDP

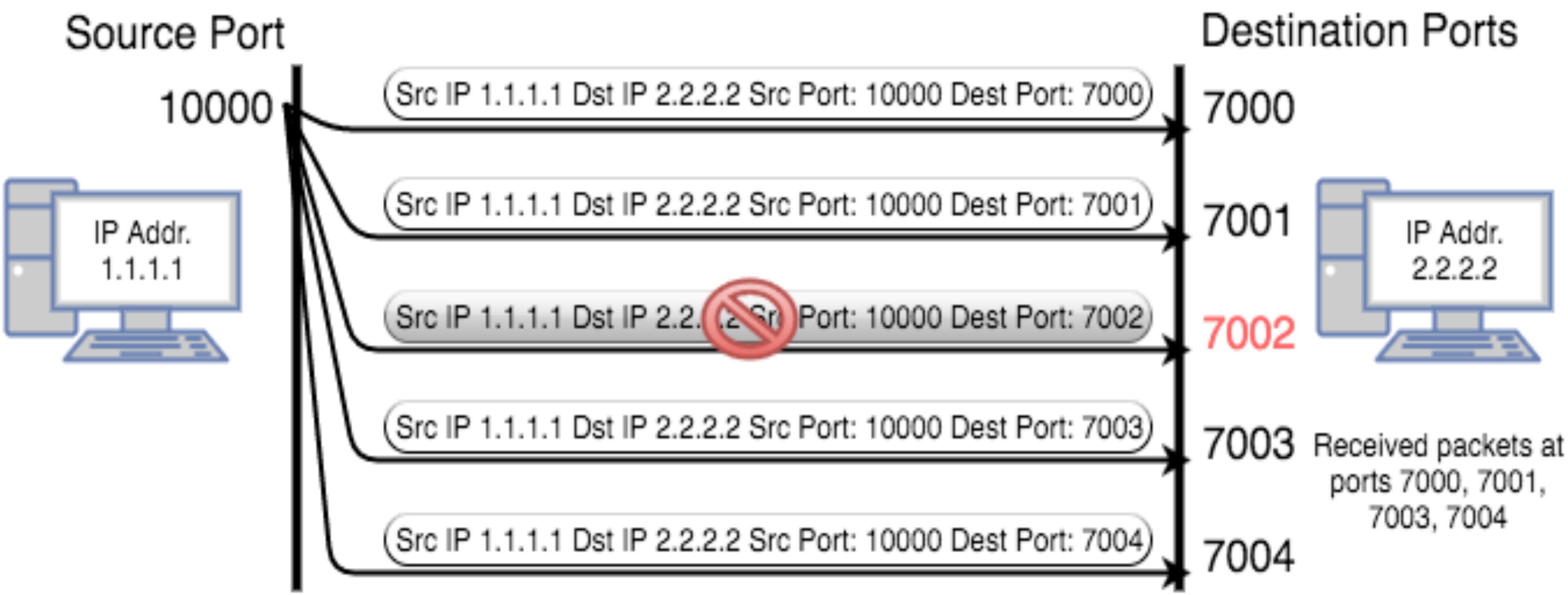
- In any normal UDP or TCP communication between two apps (or two machines), the data flows from one port-IP address pair to another port-IP address pair.



Keyed-UDP

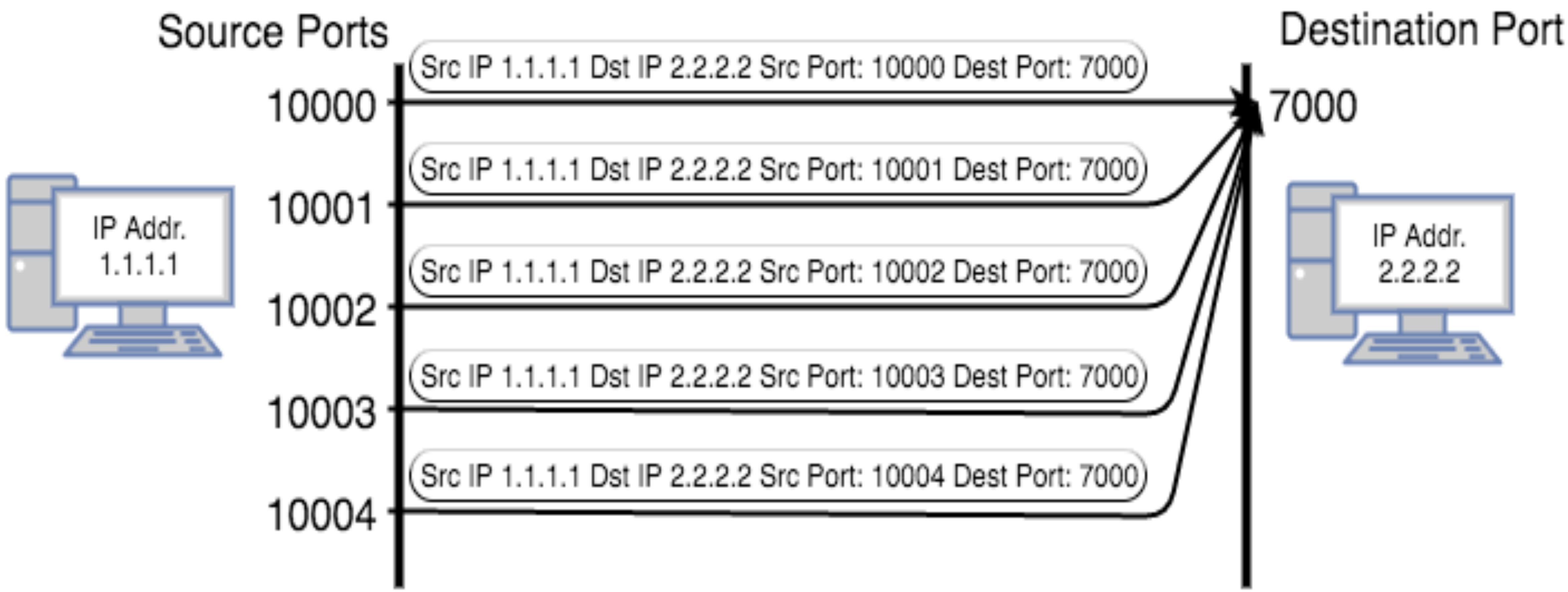

Source Port
Destination Ports
10000
IP Addr.
1.1.1.1
IP Addr.
2.2.2.2
Src IP 1.1.1.1 Dst IP 2.2.2.2 Src Port: 10000 Dest Port: 7000
Src IP 1.1.1.1 Dst IP 2.2.2.2 Src Port: 10000 Dest Port: 7001
Src IP 1.1.1.1 Dst IP 2.2.2.2 Src Port: 10000 Dest Port: 7002
Src IP 1.1.1.1 Dst IP 2.2.2.2 Src Port: 10000 Dest Port: 7003
Src IP 1.1.1.1 Dst IP 2.2.2.2 Src Port: 10000 Dest Port: 7004
7000
7001
7002
7003
7004

Keyed-UDP



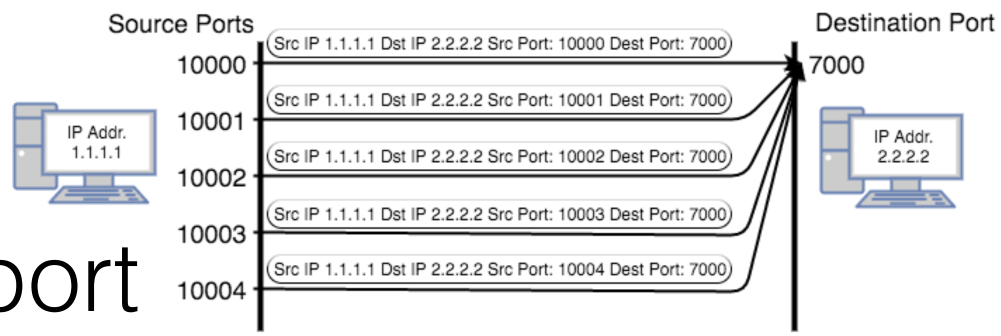
Source Port
Destination Ports
10000
Src IP 1.1.1.1 Dst IP 2.2.2.2 Src Port: 10000 Dest Port: 7000
7000
Src IP 1.1.1.1 Dst IP 2.2.2.2 Src Port: 10000 Dest Port: 7001
7001
IP Addr.
1.1.1.1
IP Addr.
2.2.2.2
Port: 10000 Dest Port: 7002
7002
Src IP 1.1.1.1 Dst IP 2.2.2.2 Src Port: 10000 Dest Port: 7003
7003
Received packets at ports 7000, 7001, 7003, 7004
Src IP 1.1.1.1 Dst IP 2.2.2.2 Src Port: 10000 Dest Port: 7004
7004

Keyed-UDP

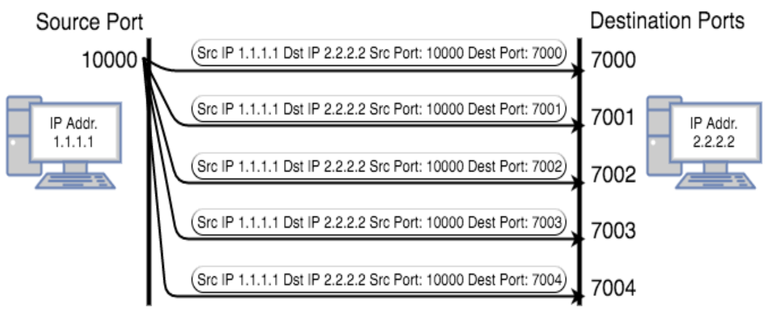


Nuno M. Garcia – ngarcia@di.ubi.pt

Keyed-UDP



- sKUDP: from many ports to one port



- dKUDP: from one port to many ports
- sdKUDP: from many ports to many ports

- Ports don't have to be sequential, *i.e.*, the key can be 7000, 5000, 7001, 7000, 5001, 7000

Keyed-UDP

- So how can the key length (and type) be detected?

Key is / Protocol	**Determined**	**Discovered**	**Agreed**
dKUDP	Hardcoded	n.a.	Key Definition Protocol
sKUDP	Hardcoded	Inference	Key Definition Protocol
sdKUDP	Hardcoded	Inference	Key Definition Protocol

Keyed-UDP

- Retro-compatibility with non-KUDP apps?

Protocol / Which application is UDP	dKUDP	sKUDP
Source application is standard UDP	Destination application receives all packets to a single destination port, disabling out-of-order and loss event detection	Destination application receives all packets from a single source port, disabling out-of-order and loss event detection
Destination application is standard UDP	Only every n^{th} packet will be received because the source application will try to send packets to destination ports that are not being monitored by the destination application	This is almost the case for the standard UDP scenario, so no loss nor out-of-order event detection is possible

Nuno M. Garcia – ngarcia@di.ubi.pt

Keyed-IPv6

- What if instead of using port numbers as keys, we use IPv6 addresses?

    - it may be feasible, most operating systems allow multiple IPv6 address assignment to an interface…

Keyed-UDP

- Pending issues:
  - What to do when the receiving app detects losses? (NOP, report back, ask to resend, data imputation - it will depend on the app)
  - Increase the complexity of the loss-switch inference algorithm by timestamping the packets at arrival?
  - Problems with non-homomorphic NAT-PT machines that are not sufficiently persistent?
  - NAT-PT tables overload? / IPv6 routing tables overload?
  - Others?

Keyed-UDP

- Proposal for a reconstruction algorithm

Rec.ved	Iteration>	1		2		3		4		5		6		7		8		9		10	
1a	1a	1a	1a		1a		1a		1a		1a		1a		1a		1a		1a		1a
3a	3a	f	3a	2b	3a		3a		3a		3a		3a		3a		3a		3a		3a
4a	4a	3a	4a	3a	4a	f	4a		4a		4a		4a		4a		4a		4a		4a
6a	6a	4a	6a	4a	6a	4a	6a	4b	6a		6a		6a		6a		6a		6a		6a
5a	5a	5a	5a	5a	5a	5a	5a	5a	5a	5a	5a		5a		5a		5a		5a		5a
2b	2b	6a	2b	6a	2b	6a	2b	6a	2b	6b	2b	6b	2b		2b		2b		2b		2b
4b	4b		4b		4b	f	4b	f	4b	f	4b	f	4b	f	4b		4b		4b		4b
6b	6b		6b		6b	2b	6b	2b	6b	f	6b	2b	6b	2c	6b	2c	6b		6b		6b
3b	3b		3b		3b	f	3b	f	3b	3b	3b	3b	3b	3c	3b	3b	3b	3b	3b		3b
2c	2c		2c		2c	4b	2c	f	2c	4b	2c	4b	2c	4b	2c	f	2c	4c	2c	4c	2c
3c	3c		3c		3c		3c	f	3c		3c	f	3c	f	3c	f	3c	f	3c	f	3c
1c	1c		1c		1c		1c	6b	1c		1c	f	1c	6b	1c	6b	1c	f	1c	6c	1c
4c	4c		4c		4c		4c		4c		4c	f	4c	f	4c	1c	4c	1c	4c	1c	4c
6c	6c		6c		6c		6c		6c		6c	2c	6c	f	6c	f	6c	2c	6c	2c	6c
1d	1d		1d		1d		1d		1d		1d		1d	3c	1d	3c	1d	3c	1d	3c	1d
2d	2d		2d		2d		2d		2d		2d		2d		2d		2d		2d		2d
4d	4d		4d		4d		4d		4d		4d		4d		4d		4d		4d		4d
3d	3d		3d		3d		3d		3d		3d		3d		3d		3d		3d		3d
5d	5d		5d		5d		5d		5d		5d		5d		5d		5d		5d		5d

Nuno M. Garcia – ngarcia@di.ubi.pt

Keyed-UDP

Ts=

{1a 2a 3a 4a *5a* 6a 1b 2b *3b* 4b 5b 6b *1c* 2c 3c 4c 5c 6c
1d 2d *3d* 4d 5d 6d 1e 2e 3e 4e *5e* 6e }

Rs=

{1a 3a 4a 6a *5a* 2b 4b 6b *3b* 2c 3c *1c* 4c
6c 1d 2d 4d *3d* 5d 6d 1e 3e 4e 6e *5e* }

Nuno M. Garcia – ngarcia@di.ubi.pt

Rec.ved	Iteration>	1		2		3		4		5		6		7		8		9		10	
1a	**1a**	1a	1a		1a		1a		1a		1a		1a		1a		1a		1a		1a
3a	**3a**	f	**3a**	2b	3a		3a		3a		3a		3a		3a		3a		3a		3a
4a	**4a**	3a	**4a**	3a	**4a**	f	4a		4a		4a		4a		4a		4a		4a		4a
6a	**6a**	4a	**6a**	4a	**6a**	4a	6a	4b	6a		6a		6a		6a		6a		6a		6a
5a	**5a**	5a	**5a**	5a	**5a**	5a	5a	5a	5a	5a	5a		5a		5a		5a		5a		5a
2b	2b	6a	**2b**	6a	**2b**	6a	2b	6a	2b	6b	2b	6b	2b		2b		2b		2b		2b
4b	4b		4b		**4b**	f	4b	f	4b	f	4b	f	4b	f	4b		4b		4b		4b
6b	6b		6b		6b	2b	6b	2b	6b	f	6b	2b	6b	2c	6b	2c	6b		6b		6b
3b	3b		3b		3b	f	3b	f	3b	3b	3b	3b	3b	3c	3b	3b	3b	3b	3b		3b
2c	2c		2c		2c	4b	2c	f	2c	4b	2c	4b	2c	4b	2c	f	2c	4c	2c	4c	2c
3c	3c		3c		3c		3c	f	3c		3c	f	3c	f	3c	f	3c	f	3c	f	3c
1c	1c		1c		1c		1c	6b	1c		1c	f	1c	6b	1c	6b	1c	f	1c	6c	1c
4c	4c		4c		4c		4c		4c		4c	f	4c	f	4c	1c	4c	1c	4c	1c	4c
6c	6c		6c		6c		6c		6c		6c	2c	6c	f	6c	f	6c	2c	6c	2c	6c
1d	1d		1d		1d		1d		1d		1d		1d	3c	1d	3c	1d	3c	1d	3c	1d
2d	2d		2d		2d		2d		2d		2d		2d		2d		2d		2d		2d
4d	4d		4d		4d		4d		4d		4d		4d		4d		4d		4d		4d
3d	3d		3d		3d		3d		3d		3d		3d		3d		3d		3d		3d
5d	5d		5d		5d		5d		5d		5d		5d		5d		5d		5d		5d

Nuno M. Garcia – ngarcia@di.ubi.pt

For the 1^{st} packet there is only one candidate, this is packet *1a*, therefore, packet *1a* takes the first position.

For the 3^{rd} packet the candidates are {*3a, 3a, f*}, therefore, packet *3a* wins.

For the 2^{nd} position, the candidates are {*f, 2b*}. As there is a tie between *2b* and *f*, the first candidate wins, *i.e.*, the algorithm concludes that the 2^{nd} packet never arrived.

For the 4^{th} position, the candidates are {*4a, 4a, 4a, 4b*}, therefore, *4a* wins.

The candidates for the remaining positions (5 to 10) are, respectively:

5^{th}: {*5a, 5a, 5a, 5a, 5a*}, elected *5a*

6^{th}: {*6a, 6a, 6a, 6a, 6b, 6b*}, elected *6a*

7^{th}: {*f, f, f, f, f*}, elected *f*

8^{th}: {*2b, 2b, f, 2b, 2c, 2c*}, elected *2b*

9^{th}: {*f, f, 3b, 3b, 3c, 3b, 3b*}, elected *3b*

10^{th}: {*4b, f, 4b, 4b, 4b, f, 4c, 4c*}, elected *4b*.

Rec.ved	Iteration>	1		2		3		4		5		6		7		8		9		10	
1a	1a	1a	1a		1a		1a		1a		1a		1a		1a		1a		1a		1a
3a	3a	f	3a	2b	3a		3a		3a		3a		3a		3a		3a		3a		3a
4a	4a	3a	4a	3a	4a	f	4a		4a		4a		4a		4a		4a		4a		4a
6a	6a	4a	6a	4a	6a	4a	6a	4b	6a		6a		6a		6a		6a		6a		6a
5a	5a	5a	5a	5a	5a	5a	5a	5a	5a	5a	5a		5a		5a		5a		5a		5a
2b	2b	6a	2b	6a	2b	6a	2b	6a	2b	6b	2b	6b	2b		2b		2b		2b		2b
4b	4b		4b		4b	f	4b	f	4b	f	4b	f	4b	f	4b		4b		4b		4b
6b	6b		6b		6b	2b	6b	2b	6b	f	6b	2b	6b	2c	6b	2c	6b		6b		6b
3b	3b		3b		3b	f	3b	f	3b	3b	3b	3b	3b	3c	3b	3b	3b	3b	3b		3b
2c	2c		2c		2c	4b	2c	f	2c	4b	2c	4b	2c	4b	2c	f	2c	4c	2c	4c	2c
3c	3c		3c		3c		3c	f	3c		3c	f	3c	f	3c	f	3c	f	3c	f	3c
1c	1c		1c		1c		1c	6b	1c		1c	f	1c	6b	1c	6b	1c	f	1c	6c	1c
4c	4c		4c		4c		4c		4c		4c	f	4c	f	4c	1c	4c	1c	4c	1c	4c
6c	6c		6c		6c		6c		6c		6c	2c	6c	f	6c	f	6c	2c	6c	2c	6c
1d	1d		1d		1d		1d		1d		1d		1d	3c	1d	3c	1d	3c	1d	3c	1d
2d	2d		2d		2d		2d		2d		2d		2d		2d		2d		2d		2d
4d	4d		4d		4d		4d		4d		4d		4d		4d		4d		4d		4d
3d	3d		3d		3d		3d		3d		3d		3d		3d		3d		3d		3d
5d	5d		5d		5d		5d		5d		5d		5d		5d		5d		5d		5d

Nuno M. Garcia – ngarcia@di.ubi.pt

Ts=

{1a 2a 3a 4a *5a* 6a 1b 2b *3b* 4b 5b 6b *1c* 2c 3c 4c 5c 6c
1d 2d *3d* 4d 5d 6d 1e 2e 3e 4e *5e* 6e }

Rs=

{1a 3a 4a 6a *5a* 2b 4b 6b *3b* 2c 3c *1c* 4c
6c 1d 2d 4d *3d* 5d 6d 1e 3e 4e 6e *5e* }

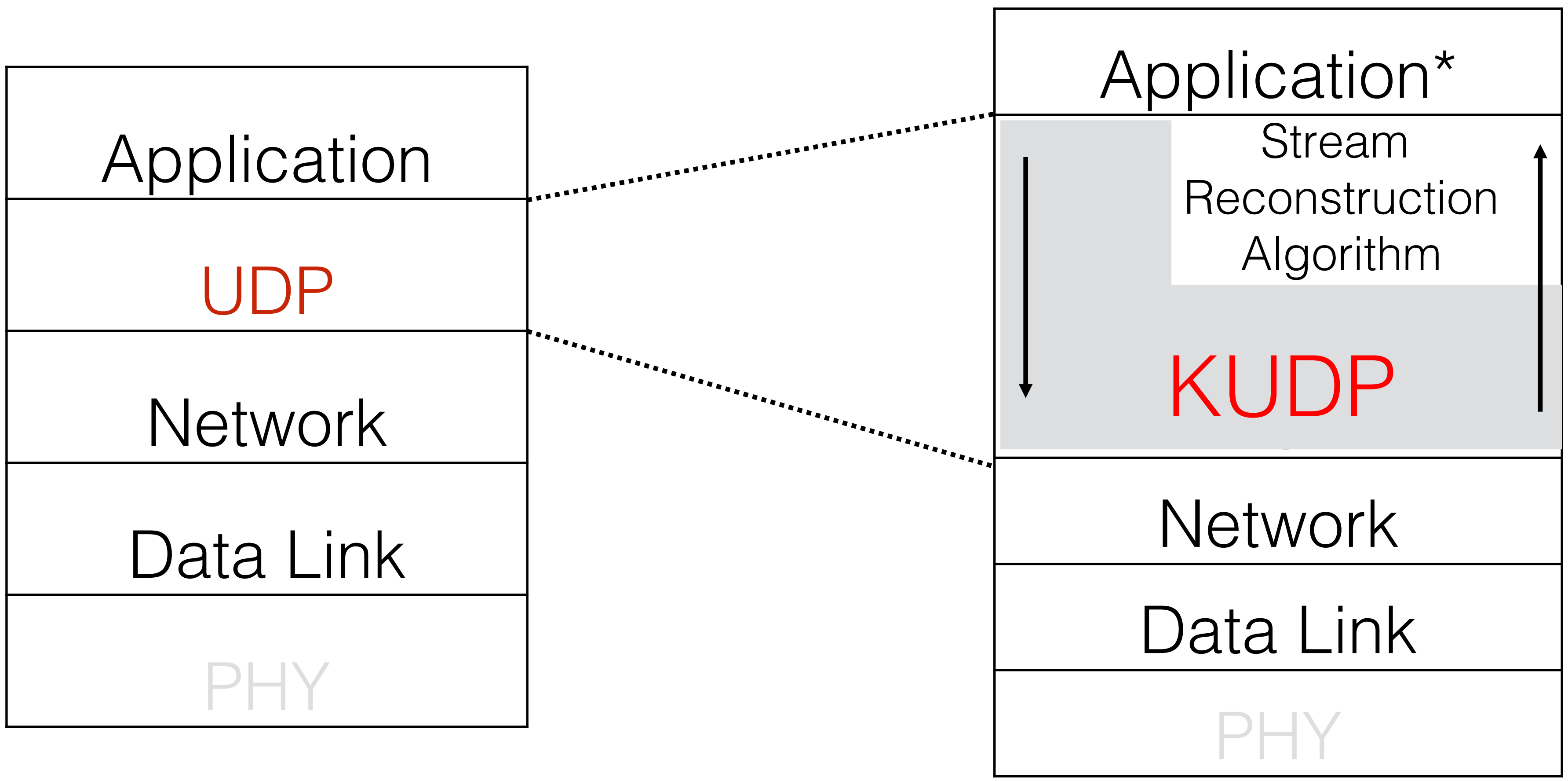
Fs=

{1a f 3a 4a 5a 6a f 2b 3b 4b f 6b 1c 2c 3c 4c 5d 6c
1d 2d 3d 4d 5d 6d 1e f 3e 4e 5e 6e }.

16.7% loss + 20% switches

Nuno M. Garcia – ngarcia@di.ubi.pt

Keyed-UDP



Nuno M. Garcia – ngarcia@di.ubi.pt

Finally, Users!

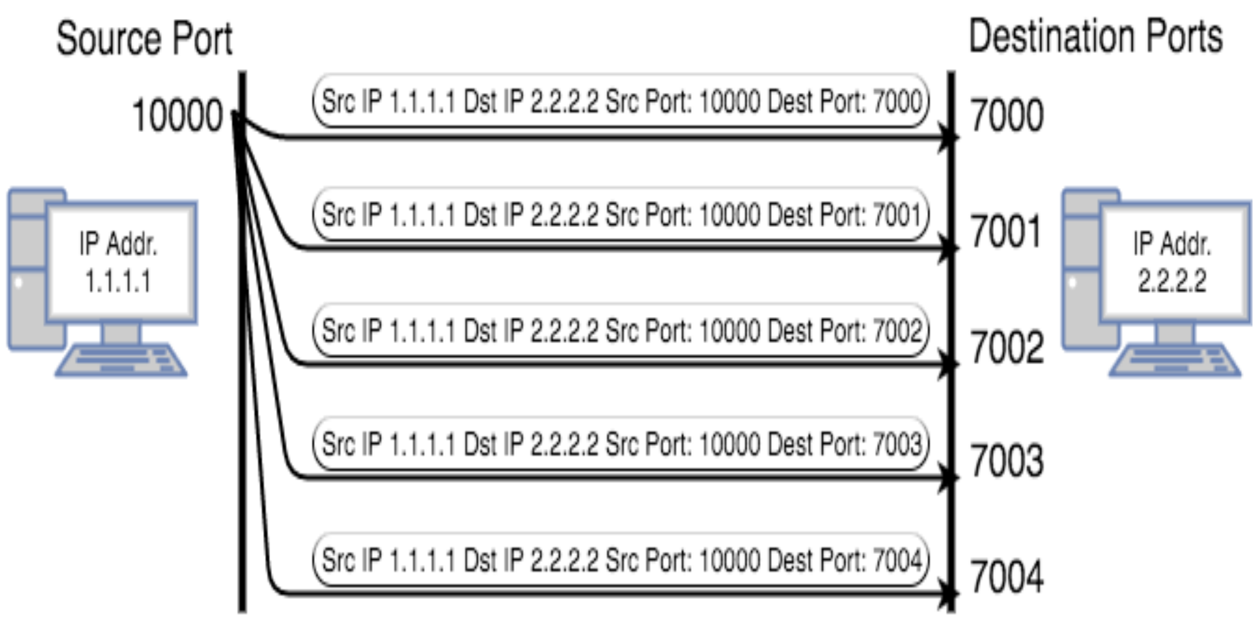
- We will be flooded with data from all of our machines
- We won't be able to process it, nor we will be aware of its existence
- We will see multi-sensor, multi-location, multi-device, multi-format data only by its impact in our daily lives (a bit like targeted advertising on the email systems)
- After the ubiquity of computers, we will have the ubiquity of data, and for users, this will mean that we will need systems that receive, summarise, process, extract relevant features and feed the appropriate algorithms for our own benefit
- The following step is to use the data from multiple users to infer and confirm daily living patterns, as to allow its monitoring and training

Fueling the Internet of Everything

- will be the Internet of our private lives, our machines, our health and our social interactions
- is this scary? a lot!
- is this worth the effort? probably yes!

Nuno M. Garcia – ngarcia@di.ubi.pt

Questions!



Thank you!

Nuno M. Garcia
Instituto de Telecomunicações
University of Beira Interior
ngarcia@di.ubi.pt

This work was supported by National Funding from the FCT - Fundação para a Ciência e a Tecnologia, through the UID/EEA/50008/2013 Project.
The author acknowledges the support of COST Action IC1303 AAPELE - Architectures, Algorithms and Protocols for Enhanced Living Environments