Time Sensitive Application Transport

Mohammad Shafahi

University of Amsterdam
Faculty of Science

Introduction

- Time sensitive applications are becoming more and more demanded
- The demand of TSAs has introduced discussions in the network community specially the NRENS community
- The NRENS main question is "What transport technology best suites time sensitive applications?"

Research Question(s)

- What are the sources of delay and jitter in technologies?
- What are the magnitudes of each source of delay and jitter?
- What are the requirements for measuring delay and jitter?
- What are the suitable methods and tools for measuring delay and jitter?

What are TSAs?

Jitter requirements of TSAs

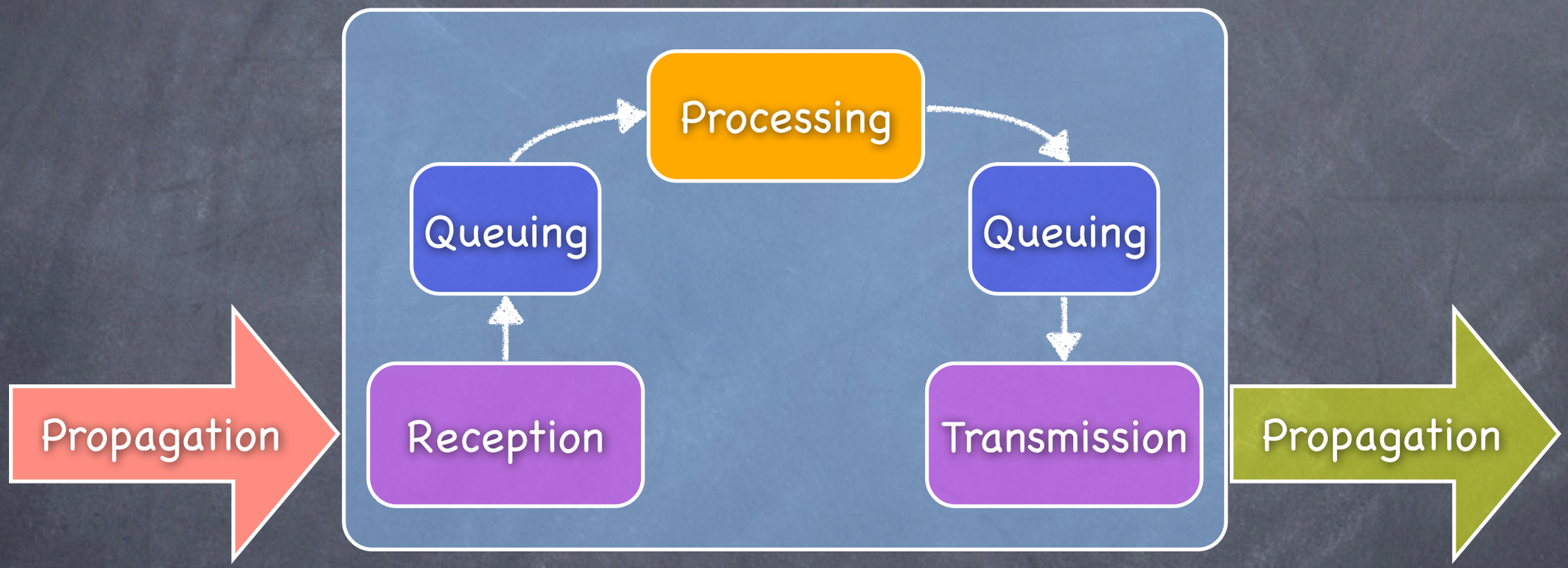
Video	Audio	Application	jitter
✔		Overlay image	240ms
✔	✔	lip synchronization	80ms
✔	✔	Music Script	5ms
	✔	Dialog	120ms
	✔	Background music	500ms
	✔	Stereo	5μs

What are the sources of delay and jitter in technologies?

Sources of delay in a network node

$$dealy_t^i = RD_t + QD1_t + PCD_t + QD2_t + TD_t + PPD_t$$

Sources of delay in a network node



$$dealy_t^i = RD_t + QD1_t + PCD_t + QD2_t + TD_t + PPD_t$$

Delay in a network path

$$delay_t =$$

$$\sum_{i=1}^{N}(RD_t^i + QD1_t^i + PCD_t^i + QD2_t^i + TD_t^i + PPD_t^i) - PD_t^N$$

Delay in a network path

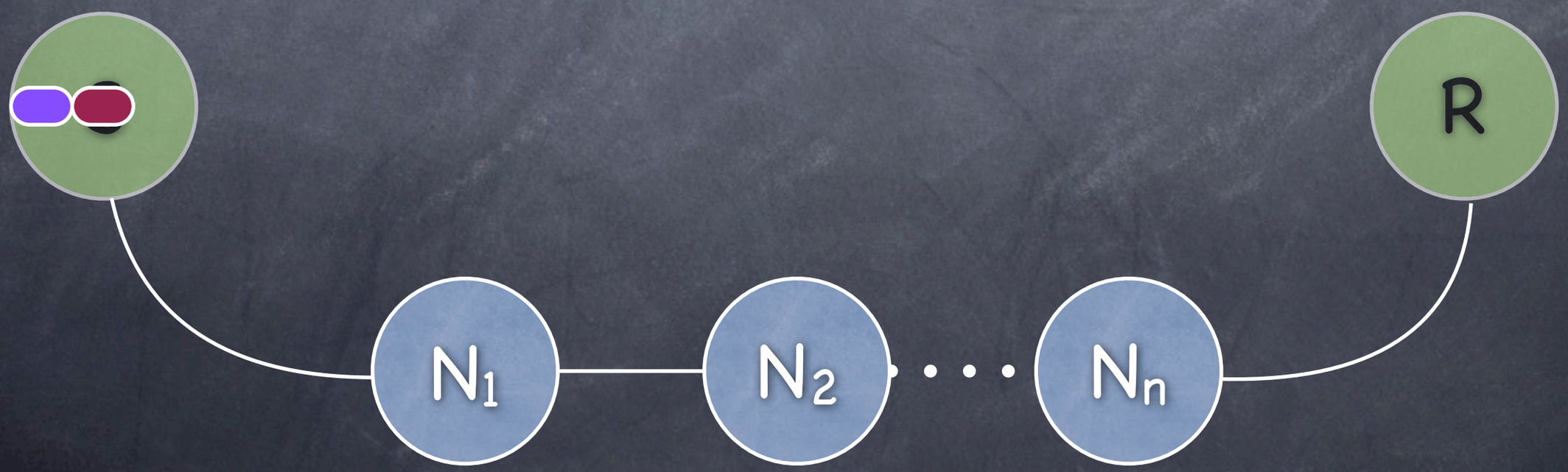
$$delay_t =$$

$$\sum_{i=1}^{N}(RD_t^i + QD1_t^i + PCD_t^i + QD2_t^i + TD_t^i + PPD_t^i) - PD_t^N$$

Definition of jitter

$$Jitter_i = delay_i - delay_{i-1}$$

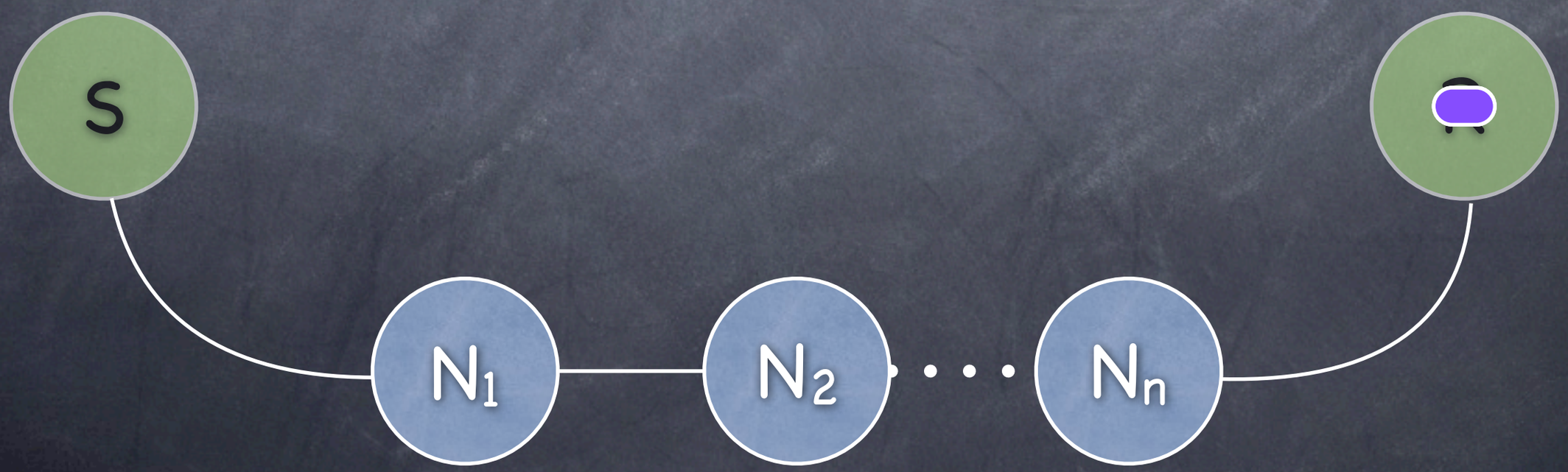
$$Jitter(\delta t) = \frac{\sum_{i=1}^{N} |jitter_i|}{N} \quad (1)$$



Definition of jitter

$$Jitter_i = delay_i - delay_{i-1}$$

$$Jitter(\delta t) = \frac{\sum_{i=1}^{N} |jitter_i|}{N} \quad (1)$$



Sources of delay

Tech/Delay	Propagation	Transmission/ Reception	Processing	Queuing
TDM		✔		
(D)WDM		✔		
SONET/SDH		✔		
OTN		✔		
Openflow			✔	
MPLS-TP			✔	
PBB-TE			✔	
Optical	✔			
Diffserv				✔

What are the magnitudes of each source of delay and jitter?

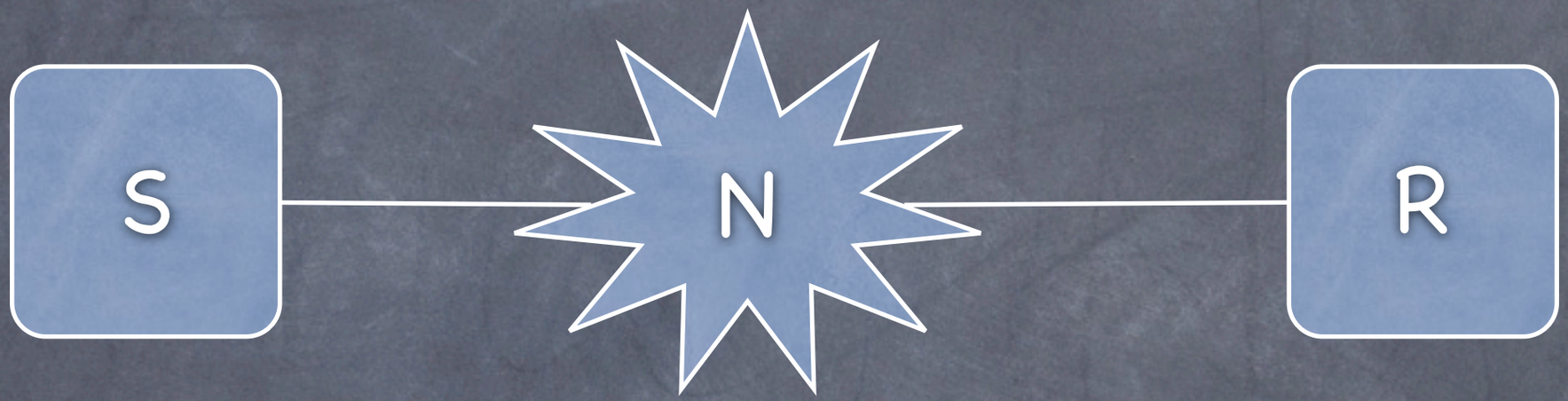
Magnitudes of delay

Source of Delay	Magnitude
Propagation	4.9 μs/km
Transmission	0.01 μs/kb*
Processing	2 μs/kb**
Queuing	$\frac{\text{Used Memory}}{\text{Service Speed}}$

* For a 100Gb interface
** Could be much less

What are the requirements for measuring delay and jitter?

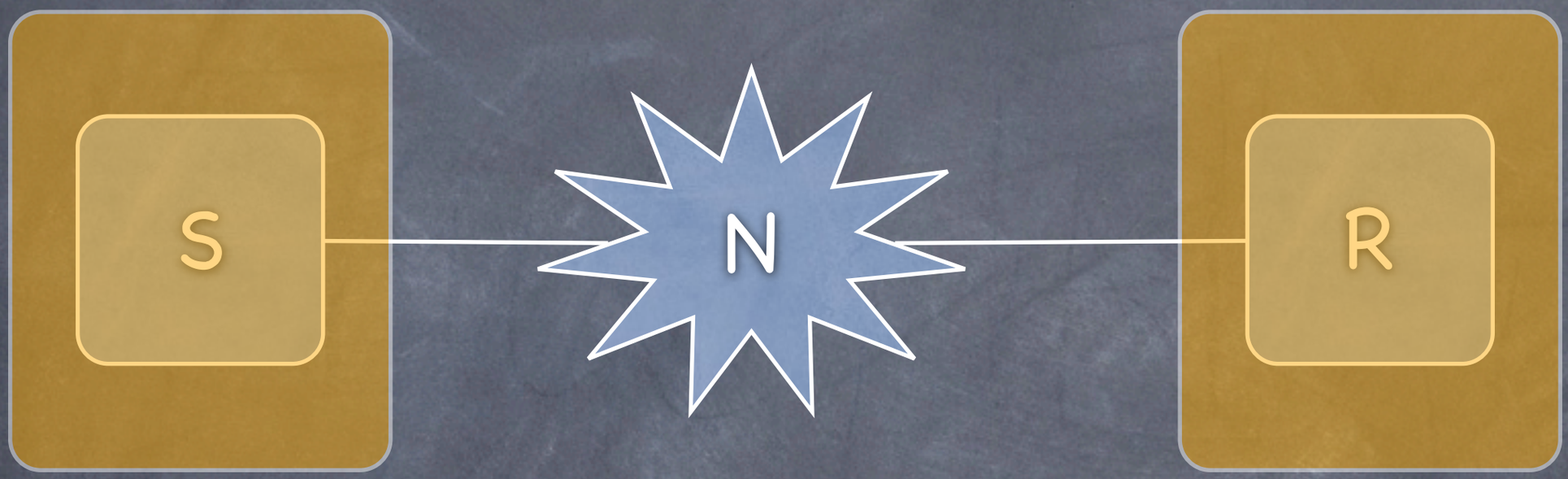
One Way Delay and jitter



T^{STS} = Send Time Registered T^{S} = Actual Send Time

T^{RTS} = Receive Time Registered T^{R} = Actual Receive Time

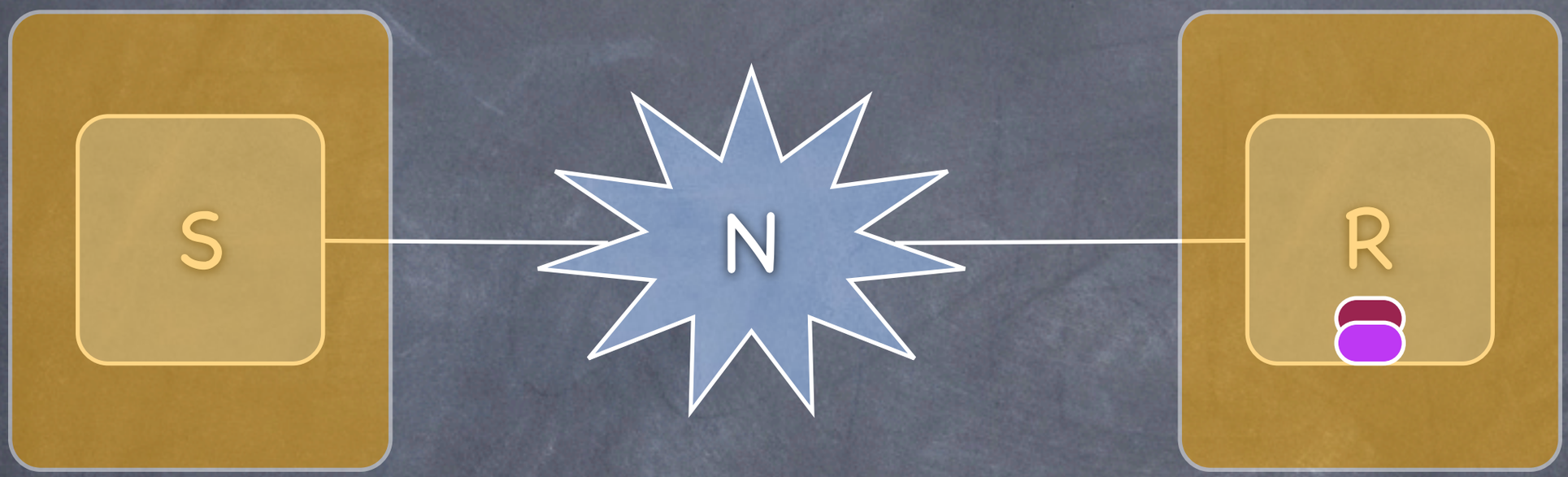
One Way Delay and jitter



T^{STS} = Send Time Registered T^{S} = Actual Send Time

T^{RTS} = Receive Time Registered T^{R} = Actual Receive Time

One Way Delay and jitter



T^{STS}_1 , T^{STS}_2

T^{S}_1 , T^{S}_2

T^{R}_1 , T^{R}_2

T^{RTS}_1 , T^{RTS}_2

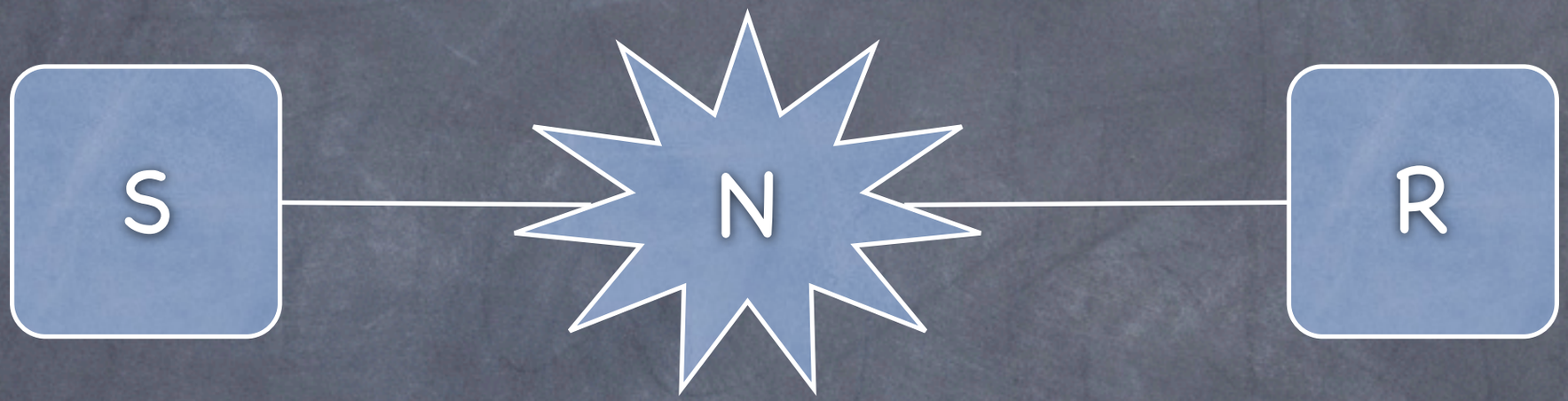
T^{STS} = Send Time Registered

T^{RTS} = Receive Time Registered

T^{S} = Actual Send Time

T^{R} = Actual Receive Time

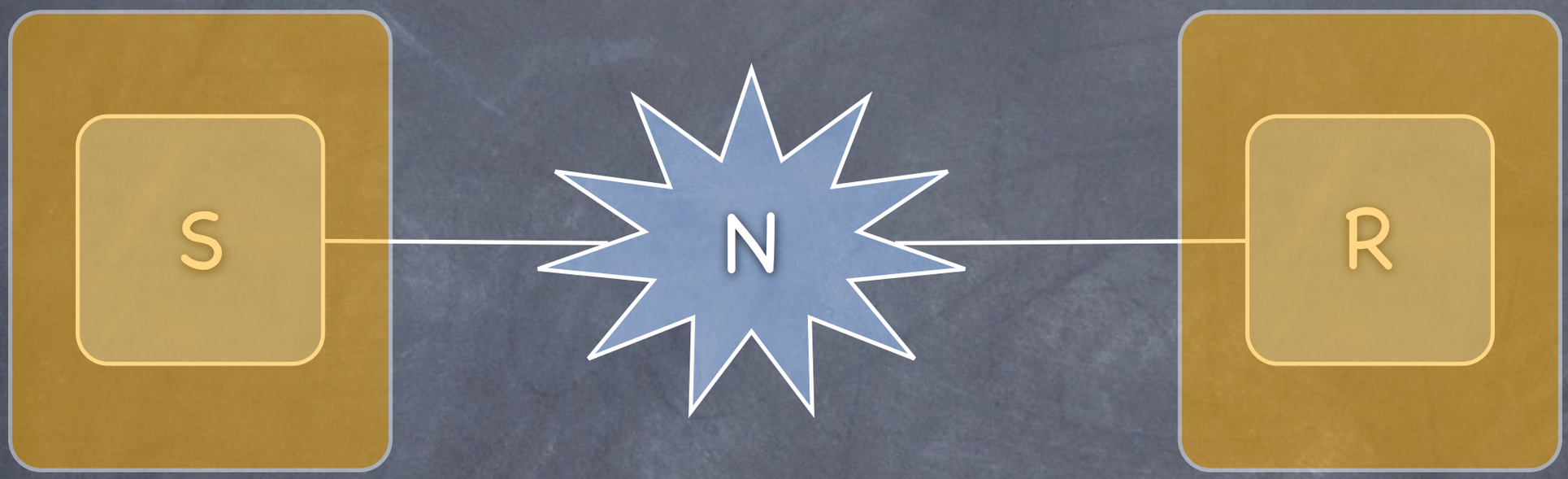
Round Trip Time and jitter



T^{STS} = Send Time Registered T^{S} = Actual Send Time

T^{RTS} = Receive Time Registered T^{R} = Actual Receive Time

Round Trip Time and jitter



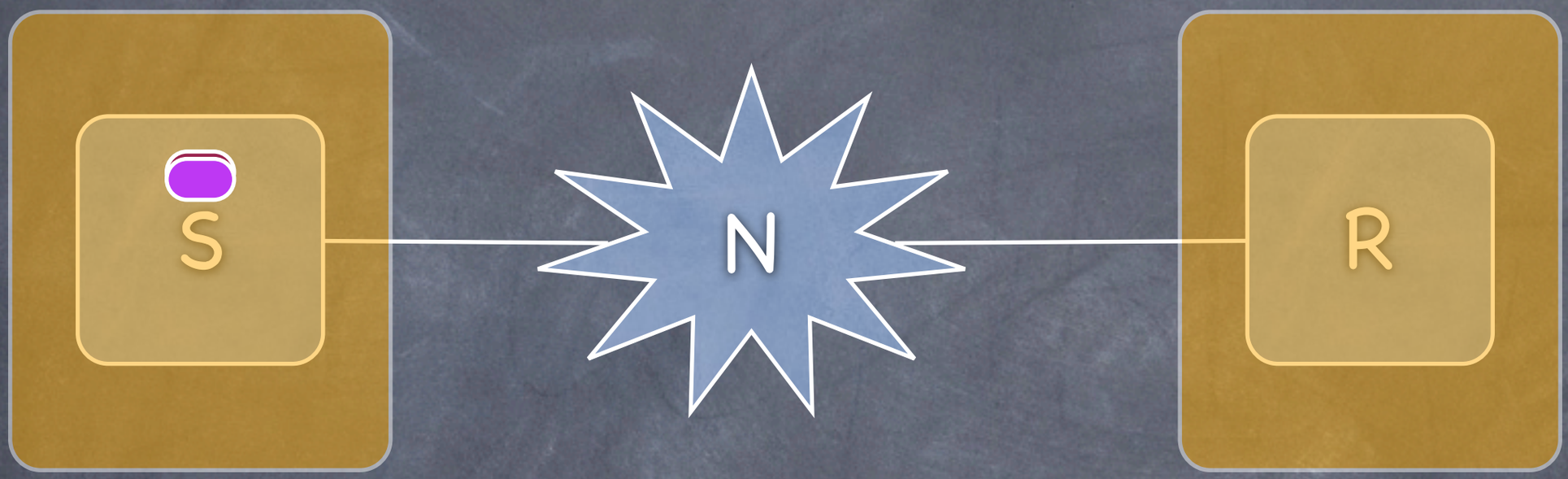
T^{STS} = Send Time Registered

T^{RTS} = Receive Time Registered

T^{S} = Actual Send Time

T^{R} = Actual Receive Time

Round Trip Time and jitter



T^{STS}_1 , T^{STS}_2

T^{S}_1 , T^{S}_2

T^{R}_1 , T^{R}_2

T^{RTS}_1 , T^{RTS}_2

T^{STS} = Send Time Registered

T^{RTS} = Receive Time Registered

T^{S} = Actual Send Time

T^{R} = Actual Receive Time

Calculating delay and jitter

$$delay_i^o = delay_i^m + delay_i^n$$

$$delay_i^n = T_i^R - T_i^S$$

$$delay_i^m = T_i^S - T_i^{STS} + T_i^{RTS} - T_i^R$$

$$jitter_i^o = delay_i^o - delay_{i-1}^o$$

$$jitter_i^n = delay_i^n - delay_{i-1}^n$$

$$jitter_i^m = delay_i^m - delay_{i-1}^m$$

Calculating delay and jitter

$$jitter^m(\delta t) + jitter^n(\delta t) = \frac{\sum_{i=1}^{N} |jitter_i^m|}{N} + \frac{\sum_{i=1}^{N} |jitter_i^n|}{N}$$

$$= \frac{\sum_{i=1}^{N} |jitter_i^m| + |jitter_i^n|}{N}$$

$$\geq \frac{\sum_{i=1}^{N} |jitter_i^o|}{N} = Jitter^o(\delta t)$$

Calculating delay and jitter

$$delay^m(\delta t) + delay^n(\delta t) = \frac{\sum_{i=1}^{N} |delay_i^m|}{N} + \frac{\sum_{i=1}^{N} |delay_i^n|}{N}$$

$$= \frac{\sum_{i=1}^{N} |delay_i^m| + |delay_i^n|}{N}$$

$$\geq \frac{\sum_{i=1}^{N} |delay_i^o|}{N} = delay^o(\delta t)$$

Measuring Delay and jitter

- One way delay measurement (OWD)
  - requires synchronous clock for measuring delay (Using NTP + GPS)
  - Works fine without synchronous clock for jitter
- Round Trip Time measurement (RTT)
  - Assumes your out going and return path and delay are the same/equal (Use the same!)

What are the suitable methods and tools for measuring delay and jitter?

Tools for measuring delay and jitter

- Hardware Solutions
  - Expensive
  - Higher accuracy
- Software Solutions
  - Cheep and many open source
  - Lower accuracy

Tools for measurement

- Hardware Solutions
  - Custom hardware
    - Passive
    - Active
  - Network devices with OAM support
- Software
  - Libraries
  - Applications

D-ITG

- Supports OWD and RTT measurement
- Supports multiple flows at the same time
- Supports multiple senders and receivers
- Supports Custom traffic patterns
- Supports a verity of protocols

Measurement accuracy setup

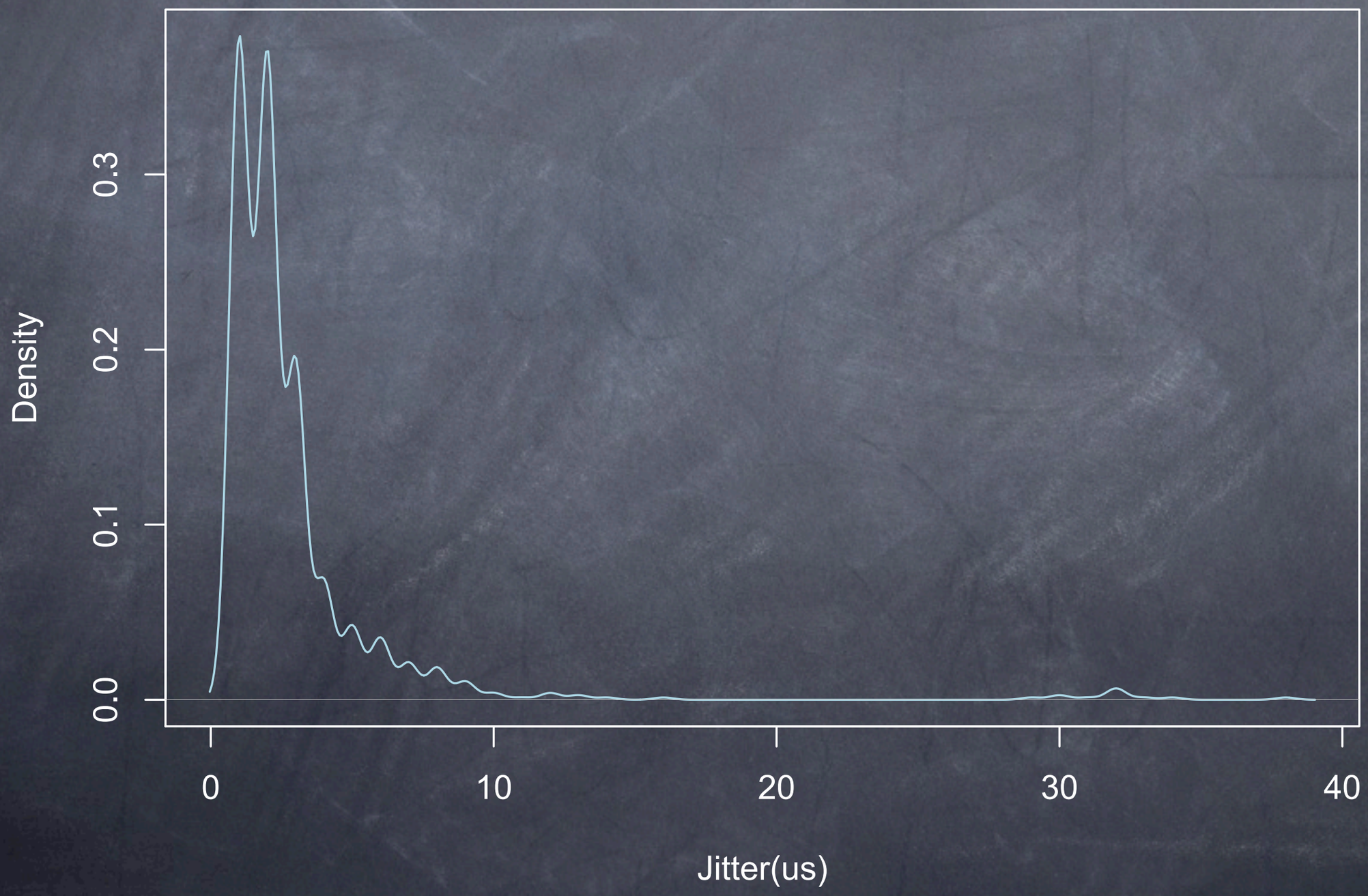
Pentium D 3GHz
2 cores
8GB ram
1 Gbit Ethernet

Cat5e

x-connect 8m

Pentium Xeon
3GHz 8 cores
16GB ram
1 Gbit Ethernet

Jitter density function of Setup



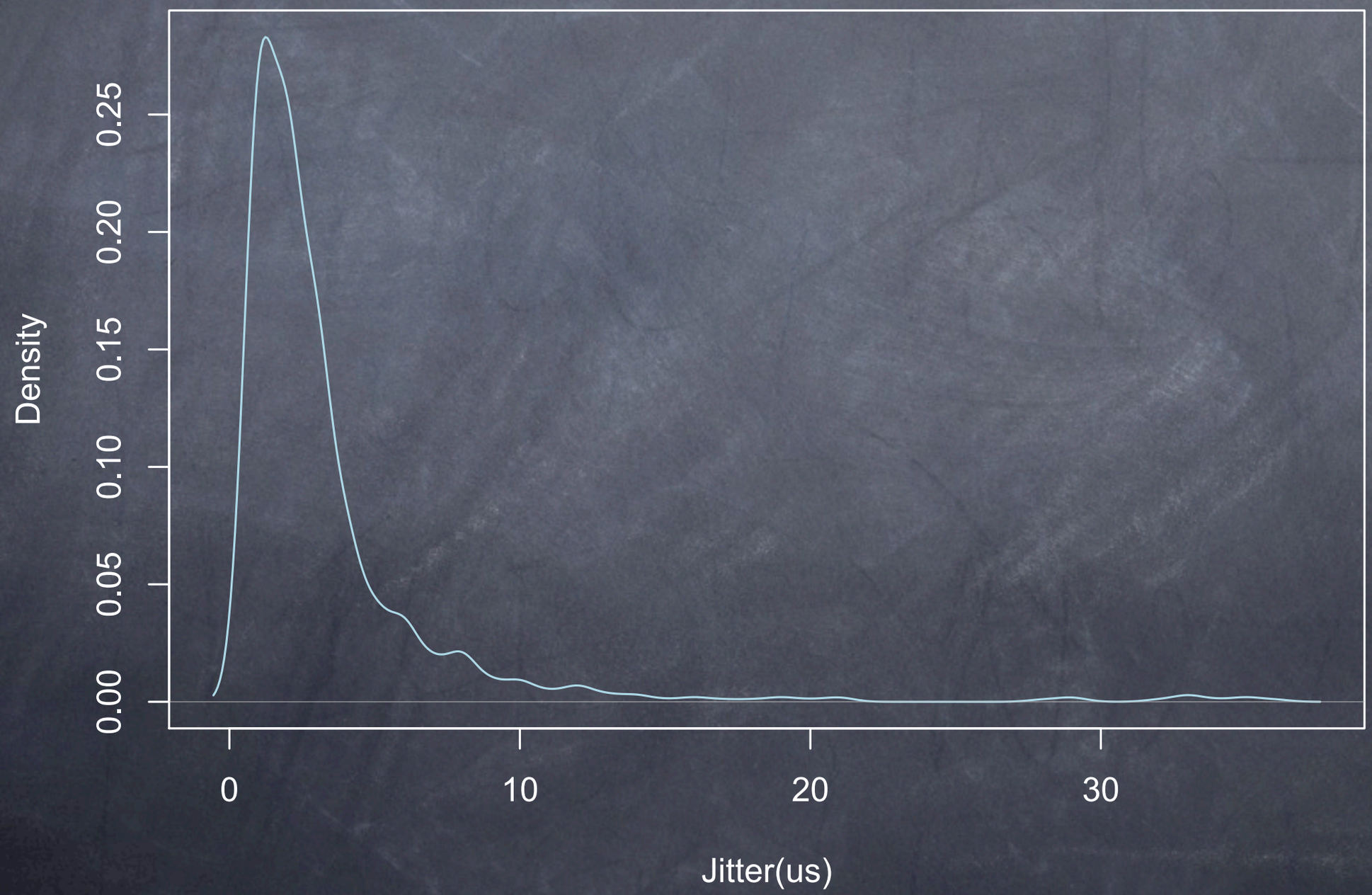
Measurement accuracy setup 2

Pentium D 3GHz
2 cores
8GB ram
1 Gbit Ethernet

Dell
PowerConnect
6248

Pentium Xeon
3GHz 8 cores
16GB ram
1 Gbit Ethernet

Jitter density function of Setup



Estimation of delay and jitter

Central Limit theorem:

The mean of a sufficiently large number of independent random variables each with finite mean and variance will approximately have a normal distribution

POC measurement setup

SUT University,
Iran

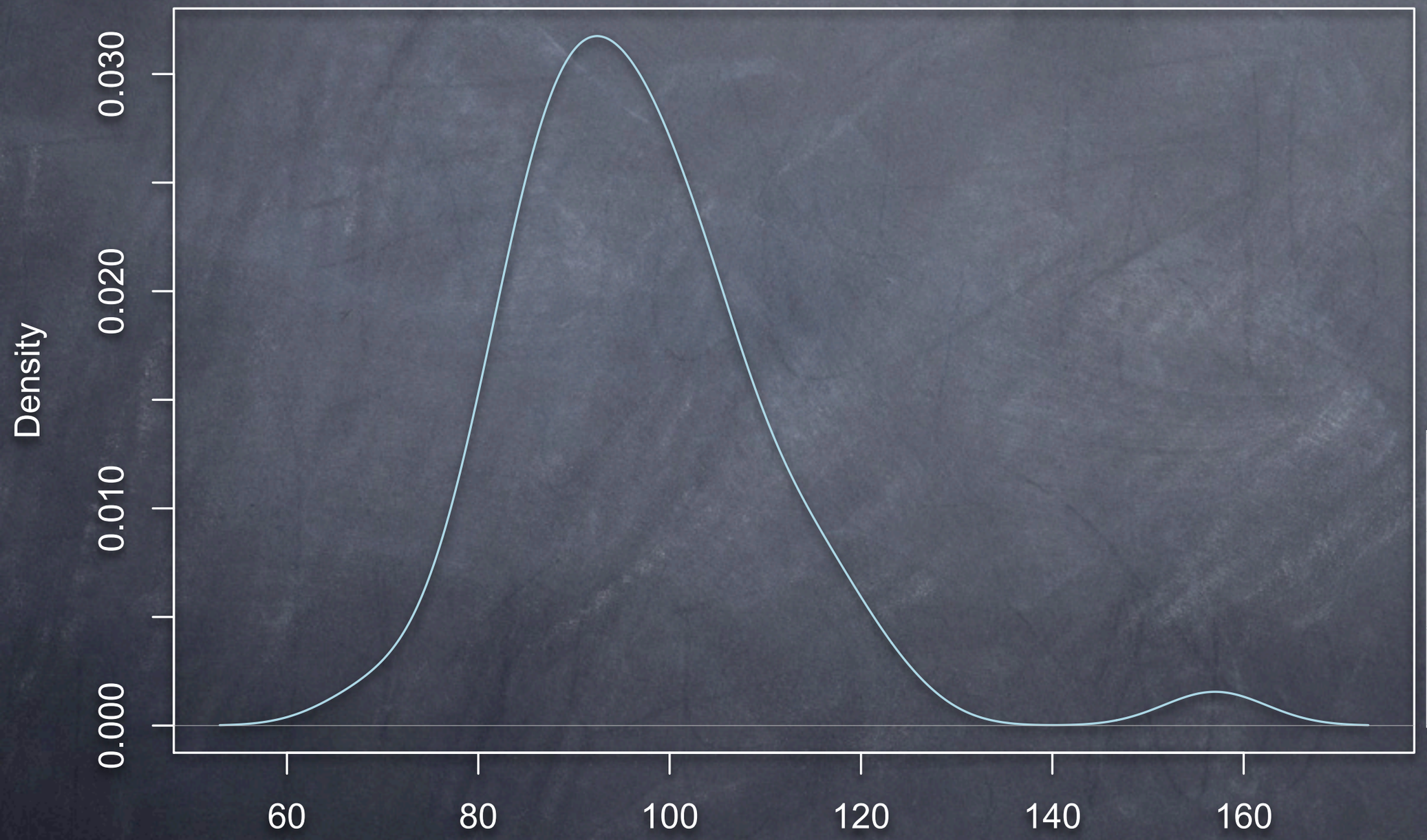
AMD Athlon 64
3.4GHz 1 cores
2GB ram
1 Gbit Ethernet

int.

UvA University,
Netherlands

Pentium Xeon
3GHz 8 cores
16GB ram
1 Gbit Ethernet

Jitter density function of setup



SD	14.3
Mean	96.8
N	12000

Conclusion

- Classified the sources of delay and provide a magnitude for them
- Provided a framework for discussion about delay and jitter
  - We have to be careful with arithmetics
- Showed that software solutions provide accuracies of microseconds in jitter measurements
- Estimate that hardware is more accurate

Further Work

- Studying hardware solutions
- Looking into different implementations of OAM in Network devices
- Adding queueing theory and system compression theories to the framework

Thanks

- Cees de laat for his great guidance
- The OS3 Research group
- My wife, fahime for her patience

Questions?

mohammad.shafahi@os3.nl
mohammad.shafahi@gmail.com