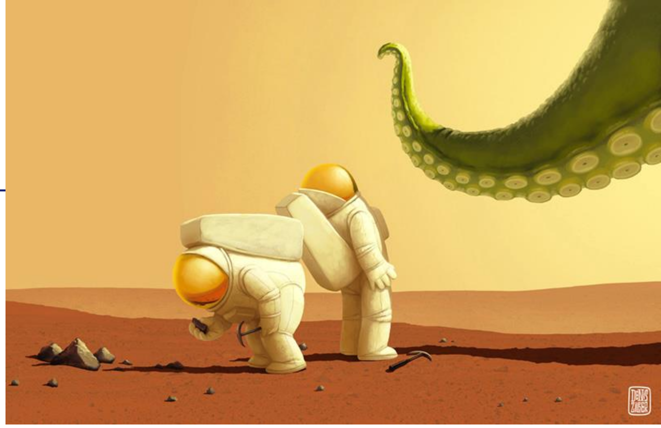
UNIVERSITÀ DEGLI STUDI
DI MILANO

Games & Network Interactivity

Lesson I06

Dario Maggiorini (dario@di.unimi.it)

Houston, We Have a Problem!



- When we play, we often complain about the quality of our online experience
  - The game is slow
  - Sorry, I got lag

    (these are only lame excuses!)

- How is this "quality of gaming experience" related to network conditions?
- What can we do to improve/measure it?

What we Usually Think

Money and sex, storage and bandwidth:
only too much is ever enough

Arno Penzias

Former Head of Bell Labs and Nobel prize winner

Network Quality of Service

- It is the capability of a network to provide better service to selected network traffic over various technologies
  - Of course, a little help is welcome
    - Circuit switching
    - Reserved resources
    - Int/Diff-serv
    - MPLS

Look this up in your undergraduate networking lecture notes

Make sure you know these when you come for the exam!

- What the heck, it's a game!
  We must serve a player not a video client streaming over DSL!

Never Underestimate Players!

300 APM → One click every 200 ms (or less)

… And Beyond

Because every input must be delivered with the right timing

... And Beyond

Because every input must be delivered with the right timing

Park Sung-Joon holds the APM world record: 818 apm!

60s / 818 = 73.3 ms

So, we **must deliver** a new move **exactly every 73.3 ms**

NOTE: making sure there is a given delay between packets IS NOT the same thing as requiring a bounded round-trip time.
Here we are talking about delivery rate only

QoS for User Experience

- Network QoS must be able to deliver an enjoyable experience for the overall feeling the player is perceiving from being in the flow
  - Read: must be good enough to make me LIKE the game despite delay and packet loss

… too bad gamers' experience is running at 400+ APM

The (Sad) Origin of QoS

- Have been born like a dream in the late ‘90
  - No definitions
  - No parameters
  - No infrastructures

Seriously, we kept talking about this for 25 years ... and all we can do today is rely on overprovisioning???

- “videoconference” become a buzzword all over the academic world, starting a big push for money and research

“To guarantee network performances to a user given some media requirements”

... so what ?

What is Network QoS (For Real) ?

- A guarantee about a data flow to have:
  - A minimum bandwidth available
  - A maximum end-to-end delay
  - A uniform inter-arrival distribution
  - A maximum rate for packet loss

… over a long-haul connection while crossing several unknown domains over the Internet

This is the only reasonable reaction, really!

"How to" Network QoS ?

- A bunch of solutions to satisfy the user's hunger for a continuous and steady data flow
  - Infrastructures
  - Algorithms
  - Strategies/Best practices
  - Protocols (?)

- Technical solutions are available
  - Int-serv (resources reservation)
  - Diff-serv (statistical guarantees)

No panic, we will just see the tip of the iceberg here

Let’s Face It!

- Network QoS never worked for games (or any other application)
  - Int-ser was not scalable enough
    - We have dozens of thousands of players connecting and disconnecting continuously
  - Statistical quality of service of diff-serv is not acceptable
    - If my bill is deterministic, then I want a deterministic QoS as well

We must work with the resources we have

The ~~.com~~ QoS bubble

- In late 2001 the bubble exploded more ore less in the same way of the .com bubble
  - Broadband availability raised
  - Bandwidth price sank
  - Companies lost interest in VoD
    - This interest resumed later, but that is another story
  - S.I.G. and W.G. (almost) disbanded
    - No one is doing research on this any more

- Result: so many cheap resources available, overprovisioning was the final answer to all questions

Luckily, I got my PhD before my advisor realized it

Overprovisioning is NOT the Answer

... It is just a trap for the fools

– Because all your telco is giving you is more bandwidth, right?

Having a huge network pipe means that we can send a lot of stuff at the same time, it does NOT mean that the stuff will reach the destination sooner!

- Example: satellite network

  available bandwidth: up to 1000 Gbps

  transmission delay (lag): average of 700 ms

Oh WOW!

Oh S#@%!

It’s a Matter of Delay!

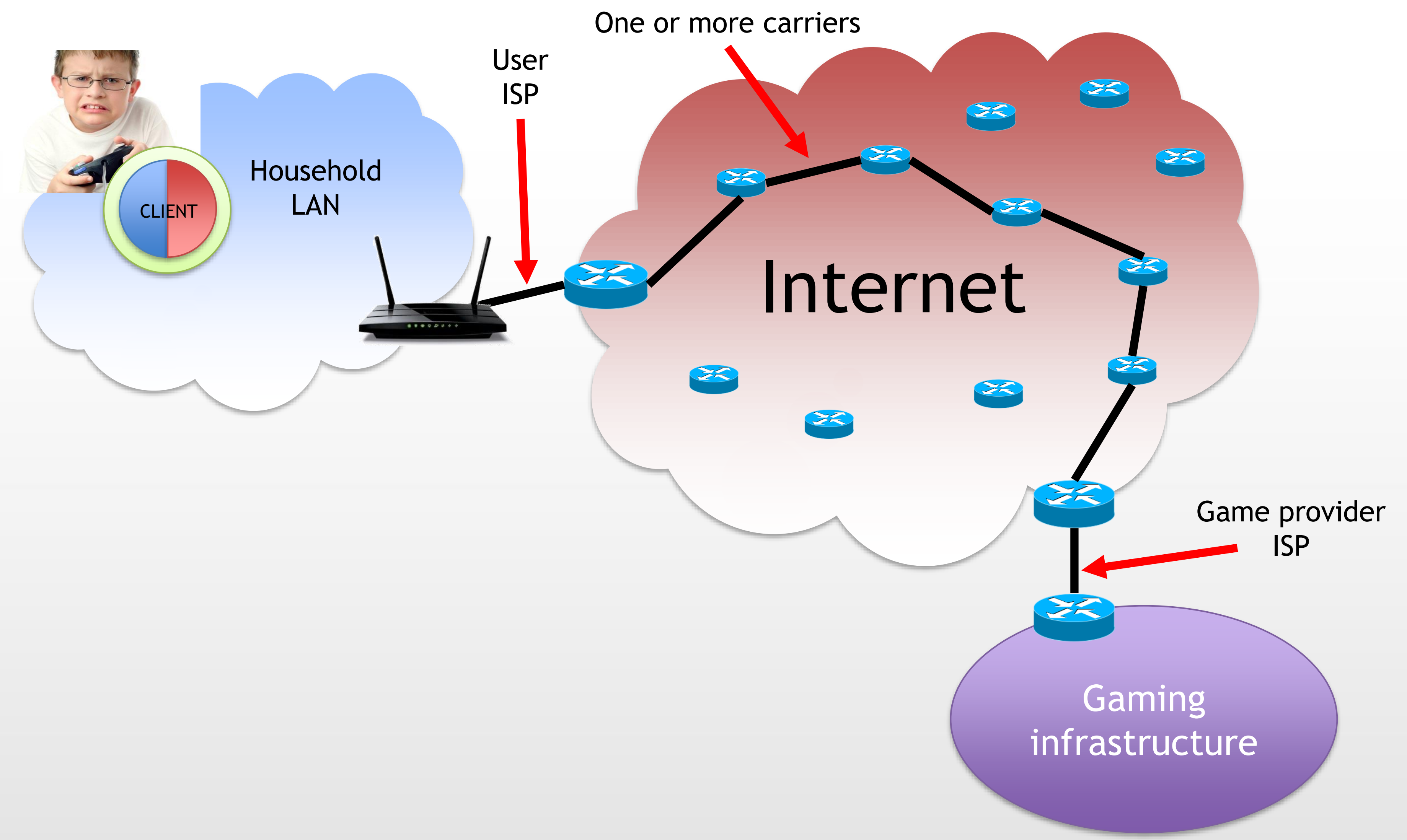
- To have data delivered in time, we must:
  - Bound transmission delay
  - Ensure jitter over the connection is close to zero

I assume you know what is jitter from a mathematical standpoint

Sorry, there is NO WAY to be sure of that on today’s networks

PRO TECH-TIP: why do I play better if I have FTTH at home? Because fiber using ATM+SONET/SDH protocols provides an isochronous transmission!

QoS Today


One or more carriers
User ISP
Household LAN
CLIENT
Internet
Game provider ISP
Gaming infrastructure

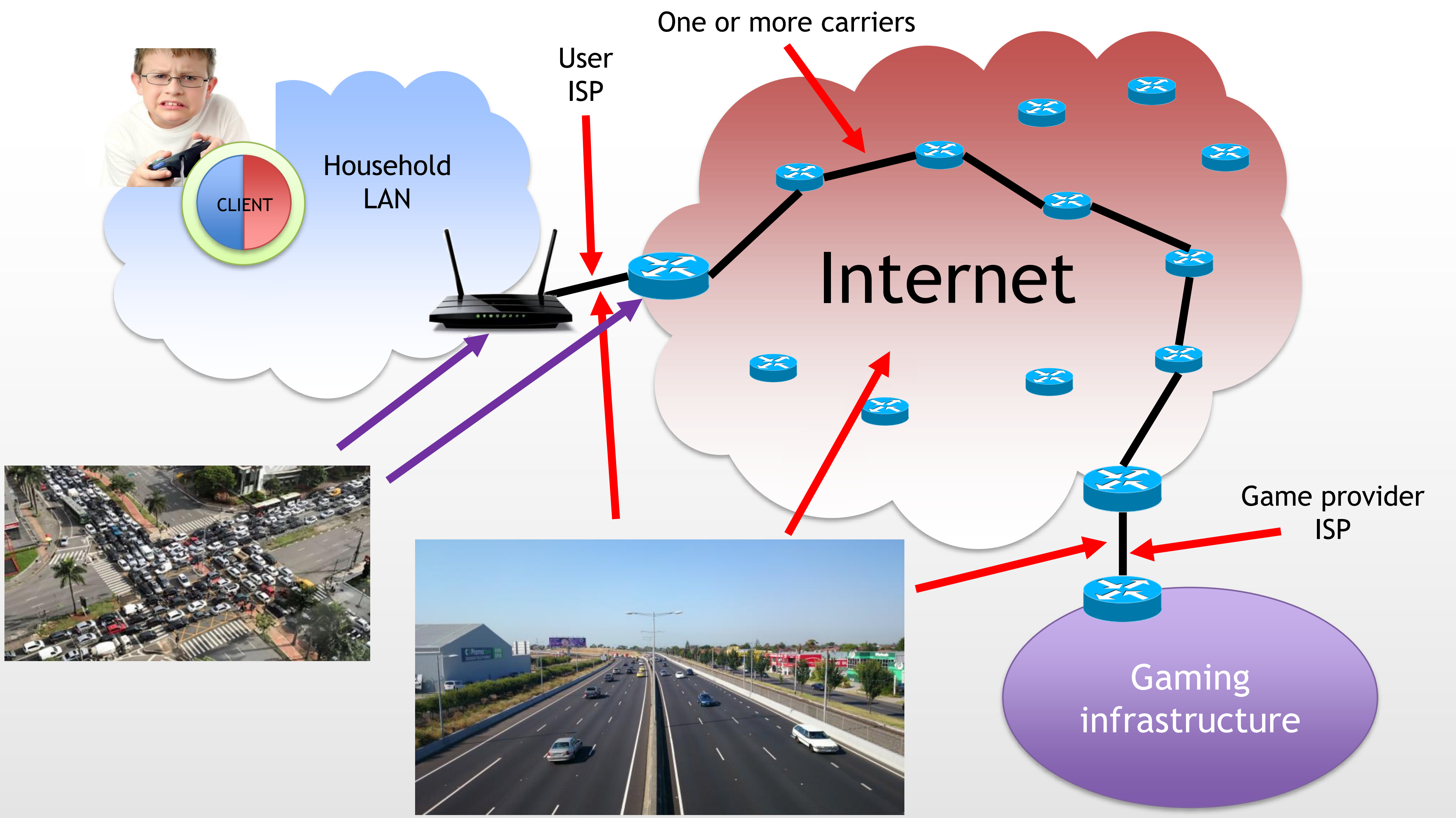
Where is the Delay Generated?

- In modern networks, delay can be generated in two ways:
  1. Congestion (in routers)
  2. Retransmission

- Retransmission is always due to packet loss (e.g., TCP) or excessive delay
- With modern network technologies it is extremely difficult to lose a packet during transmission
  - Shared channels are time slotted and ethernet is now basically point-to-point only
- As a result, we lose packets (and have retransmissions) only due to congestion

- Bottomline: if we kill the congestion, we can stop the delay

Let's Find the Congestion!



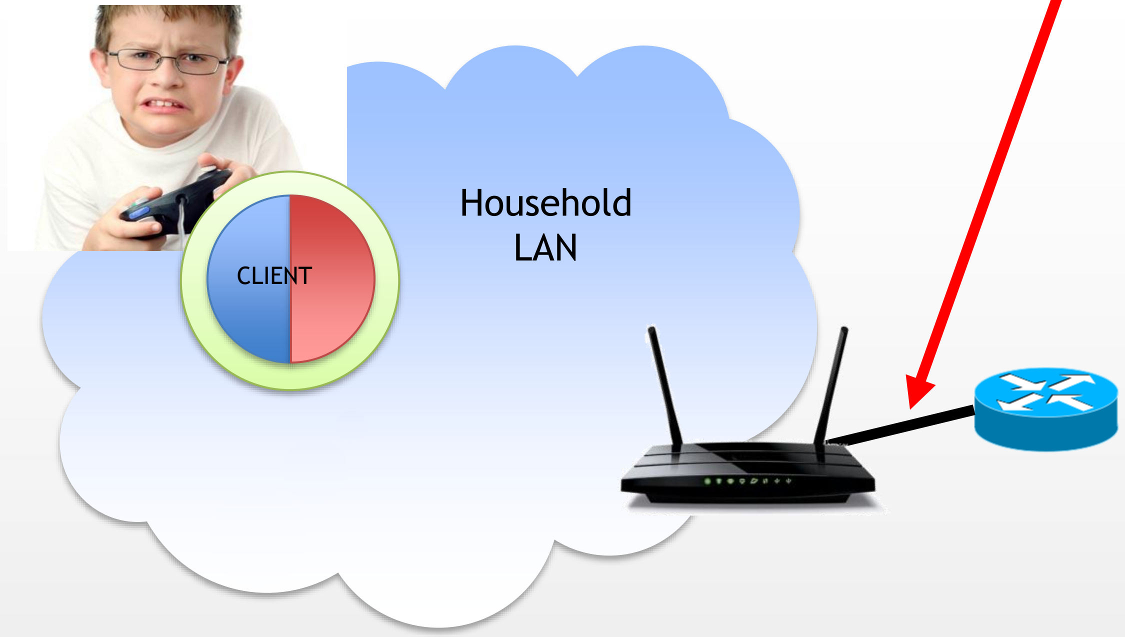
A Lot of Reasons for That!

- Too much incoming traffic from the Internet
  - Peer-to-peer (movies, music …)
  - Streaming (youtube, vimeo …)
  - Mail
- Too much outgoing traffic from the LAN
  - Peer-to-peer (movies, music …)
- Traversal traffic clogs your ISP link because there are too many active users in your area
  - this will prevent TCP acks from coming back
  - … and you retransmit after a timeout
  - … and you increase the end-to-end delay

NONE OF THIS
is related
to gaming!

The REAL Problem

Link is not letting your packets going out at the rate you need



ISP solution: PAY MORE!

Then, How do We Make People Play?

- (Easy) Solution: overprovisioning
  - Easy to deploy
    - It is already there
  - Easy to implement
    - Just pay for a bigger network pipe
  - No one should blame the game provider
    - “Ask your ISP” is in every instruction booklet and in every console help page for network malfunctions. Right?

- Congestion ?
  - A local-loop problem!

- Protocols ?
  - Applications do it better!

Let’s Play Around

- Overprovisioning
  - You must own the network
    Otherwise, the bill will be unbelievably high!
    - Owning a WAN is an IMMENSE cost; are you sure you want this?
- A last-mile problem
  - LOTS of people have med/low/crappy bandwidth at home
    If they pay to play ... It is YOUR problem!
    - Are you sure you want to lose customers?
- Applications do it better!
  - Yes ... but who is going to write the applications ?
    - YOU!

It is NOT Like That!

- Experience is a matter of quality, **not quantity**!

- Games are designed to use bandwidth sparingly
  - To save on the (provider) bill

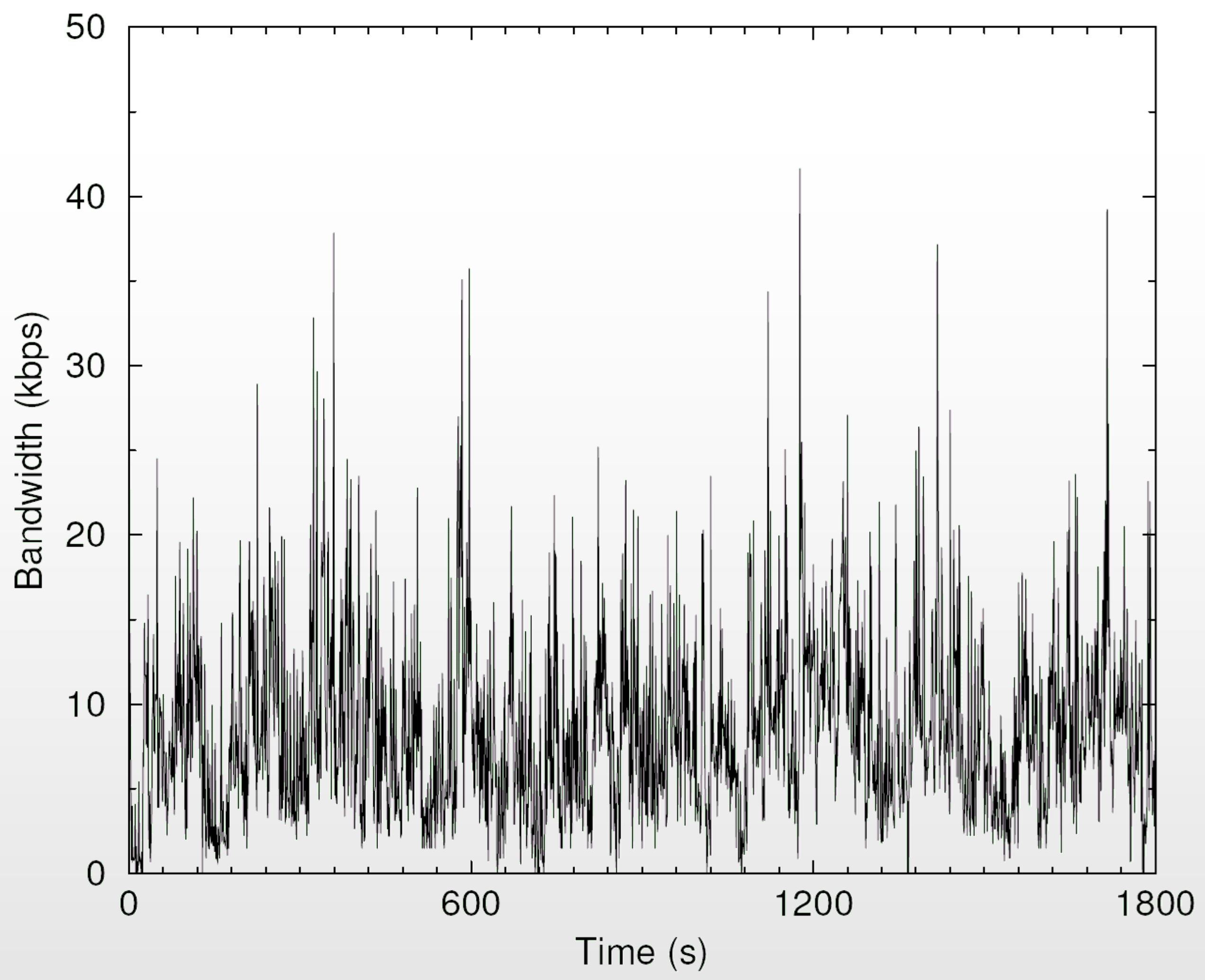
- We do not need to send much over the Internet
  - All assets are local (that is why games are so unbelievably huge)

- Ongoing research is currently focusing on three subjects
  - Understanding how network unfairness effects gameplay
  - Expanding games scalability
  - Games inter-operability

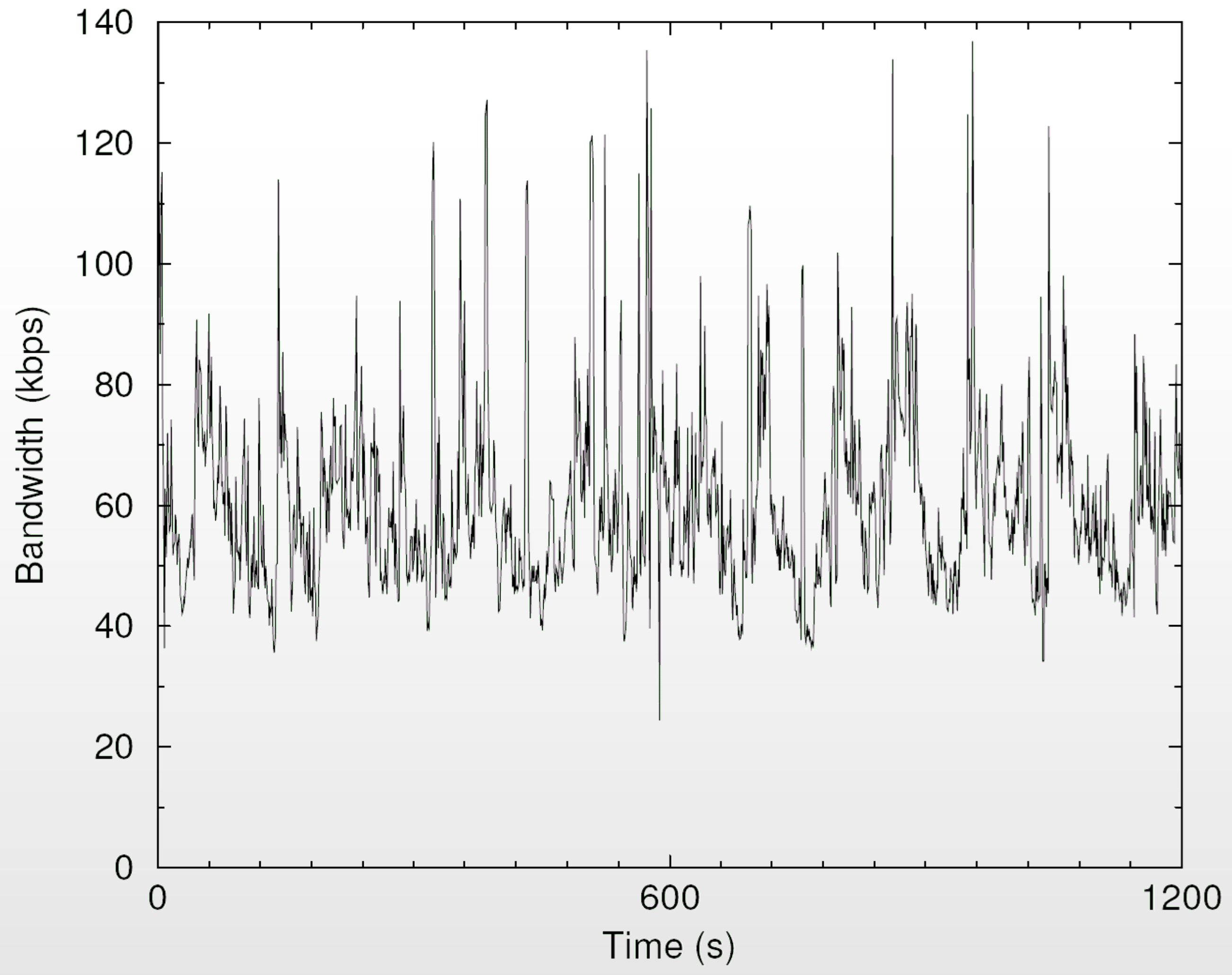
- What do we know today?

World of Warcraft Bandwidth Usage

MIND ....
KILO bps



Counter Strike Bandwidth Usage

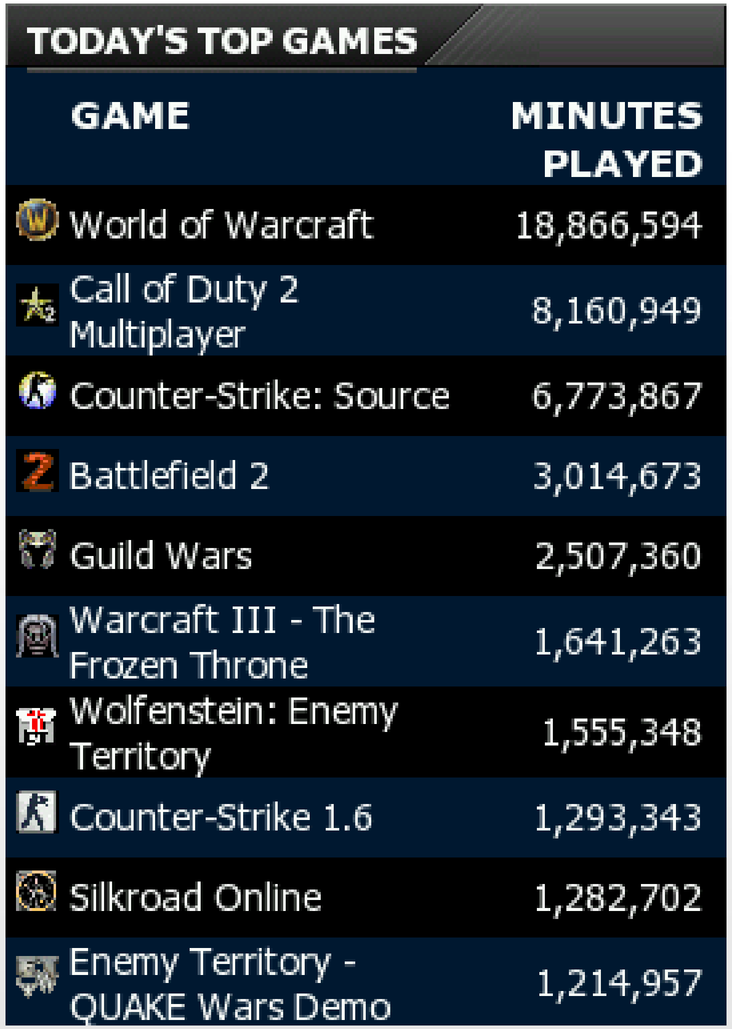


Why Saving Bandwidth

- MMORPG owners use 33% of subscription fees to pay networking and data center operations
- ... and no one like bills

VERY OLD (xfire.com is not longer online) report about online games usage

Total minutes played per month as of Sep. 2007

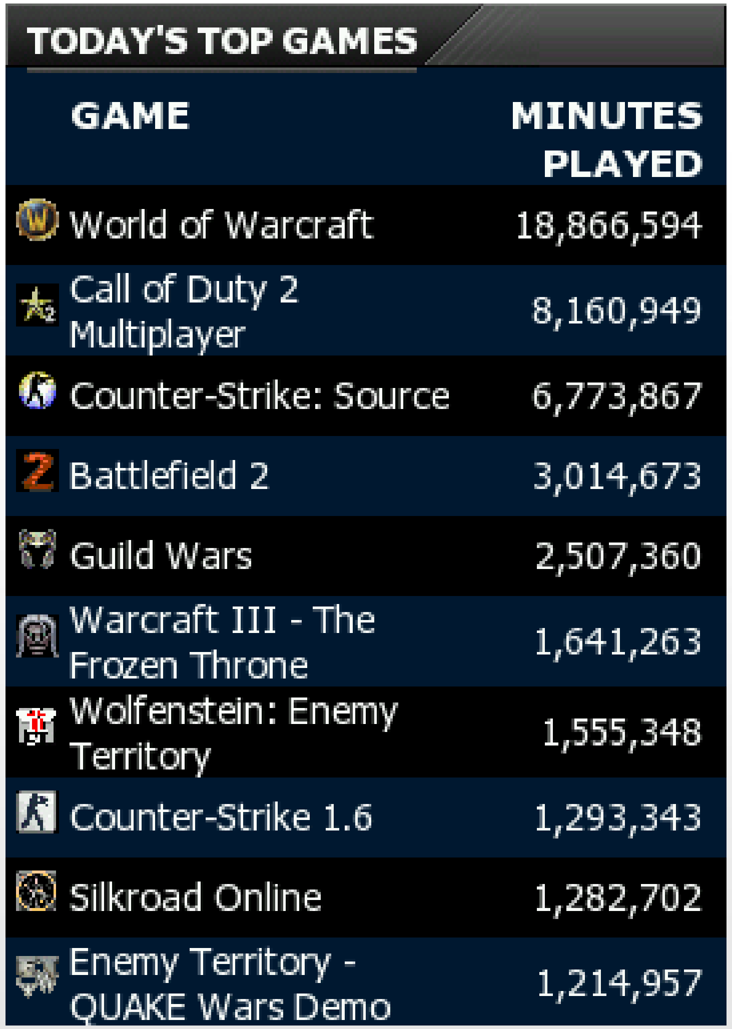


TODAY'S TOP GAMES

GAME	MINUTES PLAYED
World of Warcraft	18,866,594
Call of Duty 2 Multiplayer	8,160,949
Counter-Strike: Source	6,773,867
Battlefield 2	3,014,673
Guild Wars	2,507,360
Warcraft III - The Frozen Throne	1,641,263
Wolfenstein: Enemy Territory	1,555,348
Counter-Strike 1.6	1,293,343
Silkroad Online	1,282,702
Enemy Territory - QUAKE Wars Demo	1,214,957

Why Saving Bandwidth

- Let’s do some math on WoW:
  - 18866594 total gaming minutes
  - If all users connect at 300 kbps constantly (KILO!)
  - Total traffic per day: ~ 38 TB
  - Required CBR bandwidth (on server side): 3.6 Gbps !
- Believe me, you DO NOT want to know how much is the price of such a connection



TODAY'S TOP GAMES

GAME	MINUTES PLAYED
World of Warcraft	18,866,594
Call of Duty 2 Multiplayer	8,160,949
Counter-Strike: Source	6,773,867
Battlefield 2	3,014,673
Guild Wars	2,507,360
Warcraft III - The Frozen Throne	1,641,263
Wolfenstein: Enemy Territory	1,555,348
Counter-Strike 1.6	1,293,343
Silkroad Online	1,282,702
Enemy Territory - QUAKE Wars Demo	1,214,957

Long Term Bandwidth Usage

- This is called “patch distribution”
  - As in “download capability”
- You cannot distribute a 150+ MB patch to be downloaded
  - And maybe … every few days
  - And, even worst, on mobile metered networks
- This is not a technical problem rather than an organization problem
- Oh well, someone should tell this to AAA companies

Let’s Dive Into Latency (LAG) and Gaming

- Basically, your Round-Trip-Time PLUS calculations on server
  - We never ask to client to take decisions! More on this later in the course

- DSL – average is 60 ms
- Dial-up – average is 200 ms
- GPRS – seconds (!)
- UMTS – hundreds of ms
- LTE (4G) – hundreds of ms
- 5G - nearly nothing ... when talking to the edge

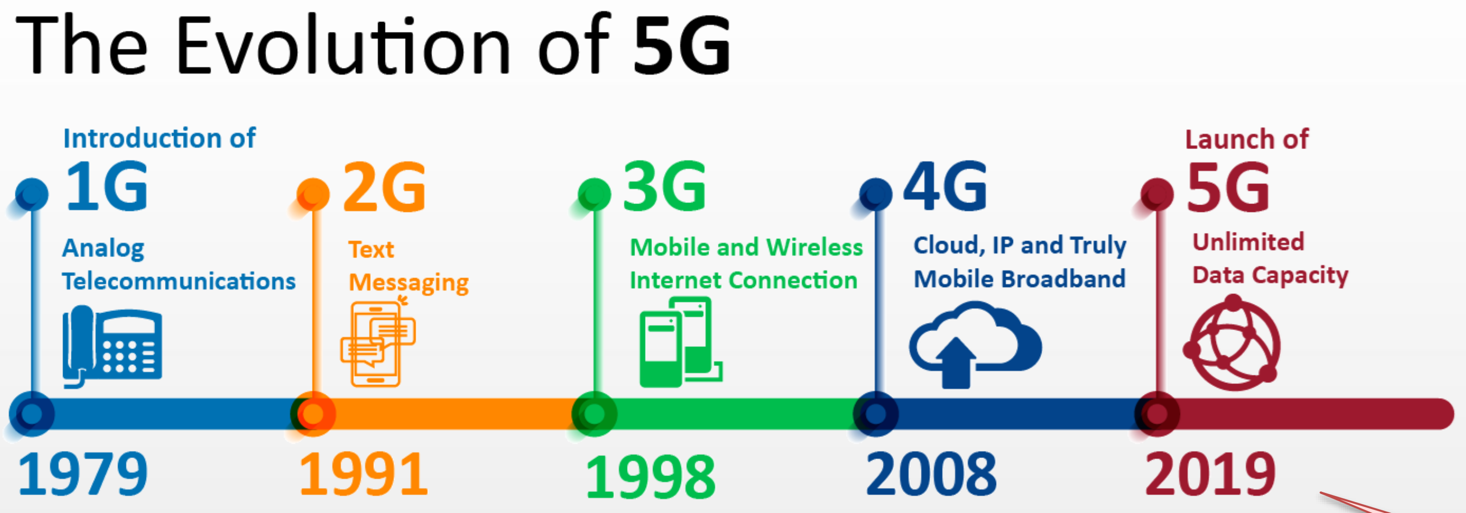
Mainly due to firewalls

Ouch!

- Think in term of number of frames (30 to 60 or more per second) and you can get the idea

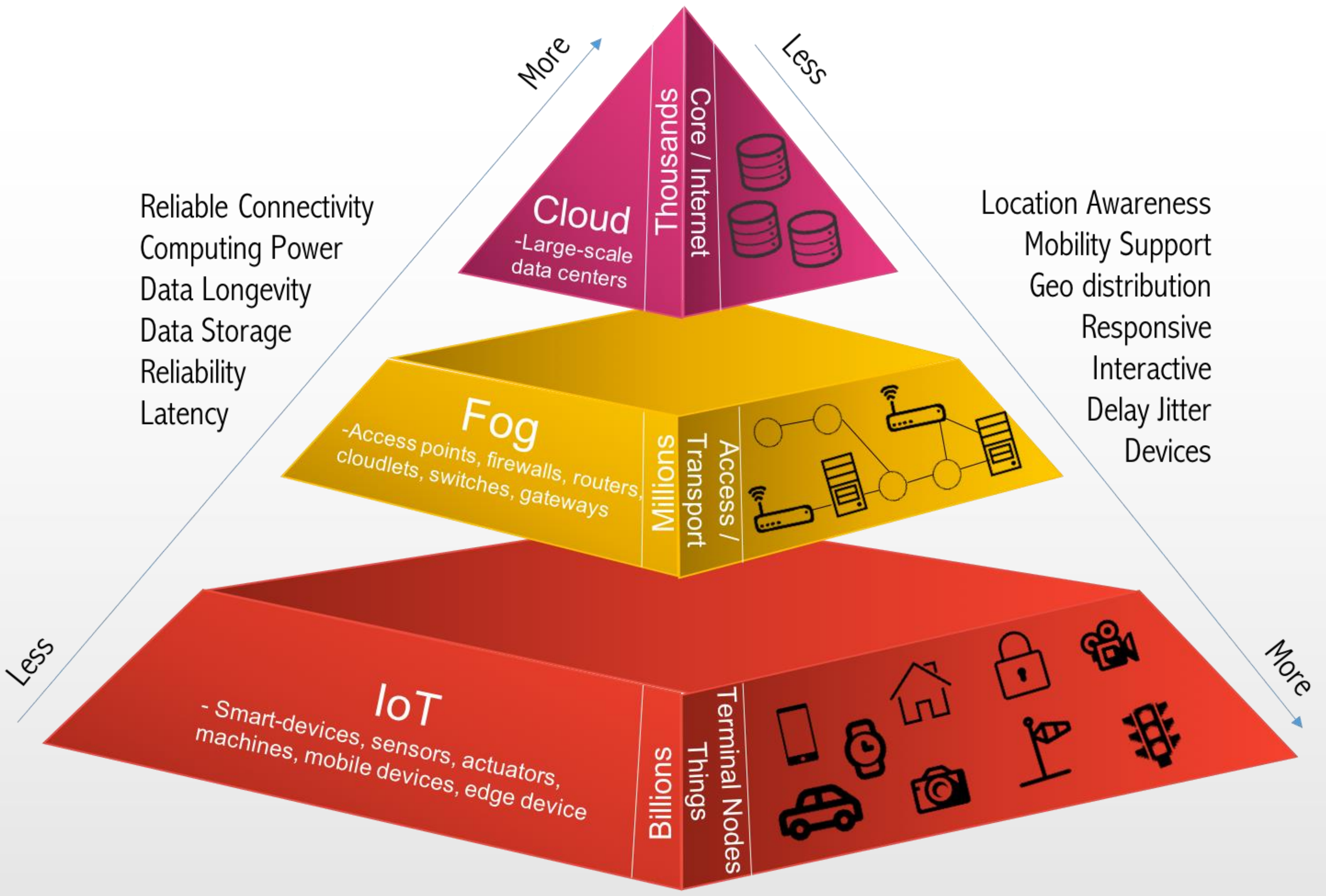
5G Networks

… and we are already lagging behind



Service anyone?

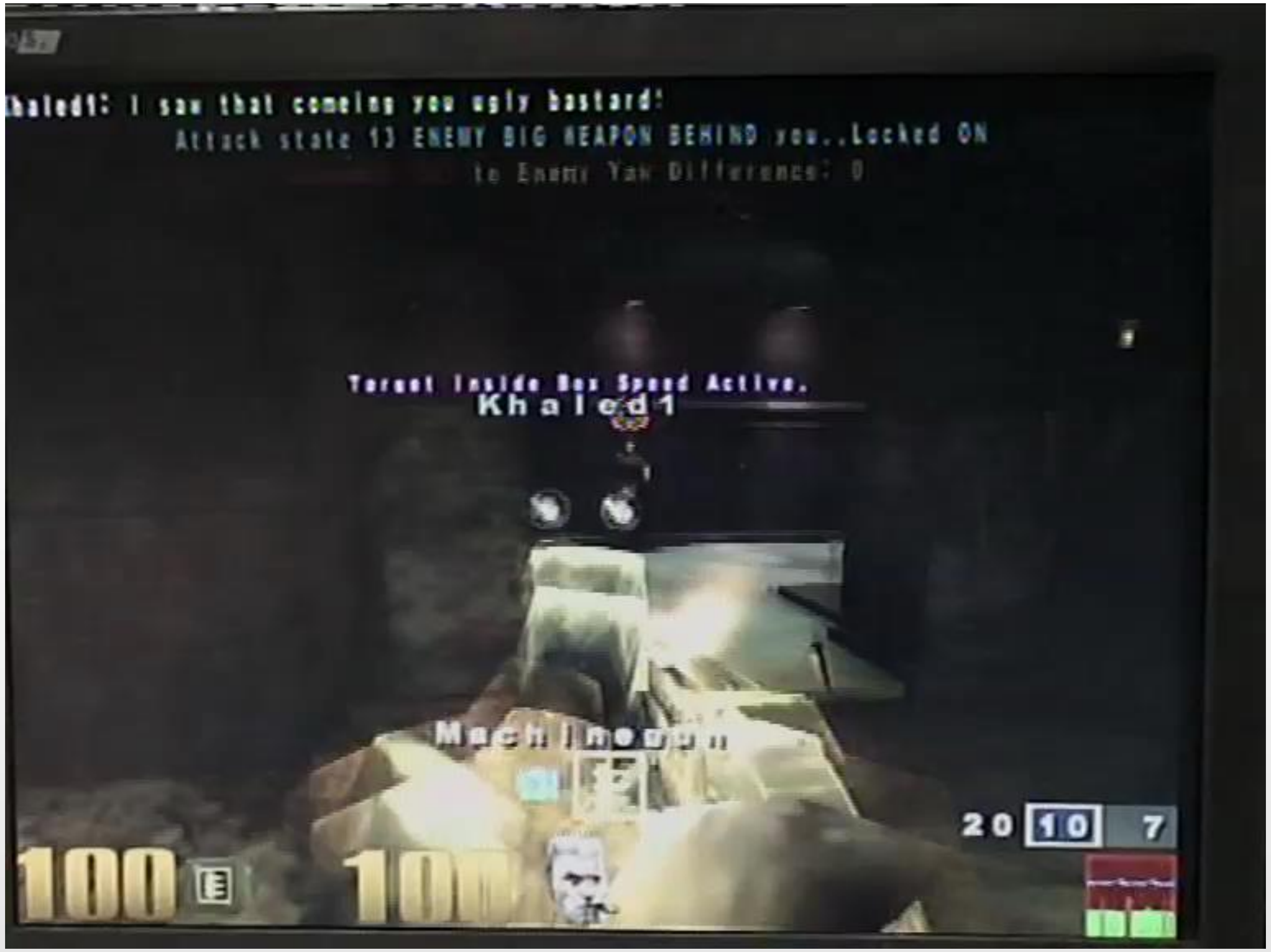
5G Networks


More
Less
Less
More
Reliable Connectivity
Computing Power
Data Longevity
Data Storage
Reliability
Latency
Location Awareness
Mobility Support
Geo distribution
Responsive
Interactive
Delay Jitter
Devices
Cloud
-Large-scale data centers
Thousands
Core / Internet
Fog
-Access points, firewalls, routers, cloudlets, switches, gateways
Millions
Transport
Access /
IoT
- Smart-devices, sensors, actuators, machines, mobile devices, edge device
Billions
Things
Terminal Nodes

How We Deal With Delay in Games

- As a matter of fact, when we play, we care only about LAG
  - Ever wandered there is no “bandwidth hog” indicator in any client?
  - Even measuring it is not very straightforward
- Usually, we feel happy (and fair) if all players suffer from the same “acceptable” delay
  - LAG compensation techniques can be applied
    - The delay of all connection is made even with the worst one
  - The definition of “acceptable” depends on the type of game
- If the LAG is (fairly) increased (too much) for everyone the game may be perceived as playable, but we might see some nasty side-effects

Lag Compensation is NOT Helping in All Cases



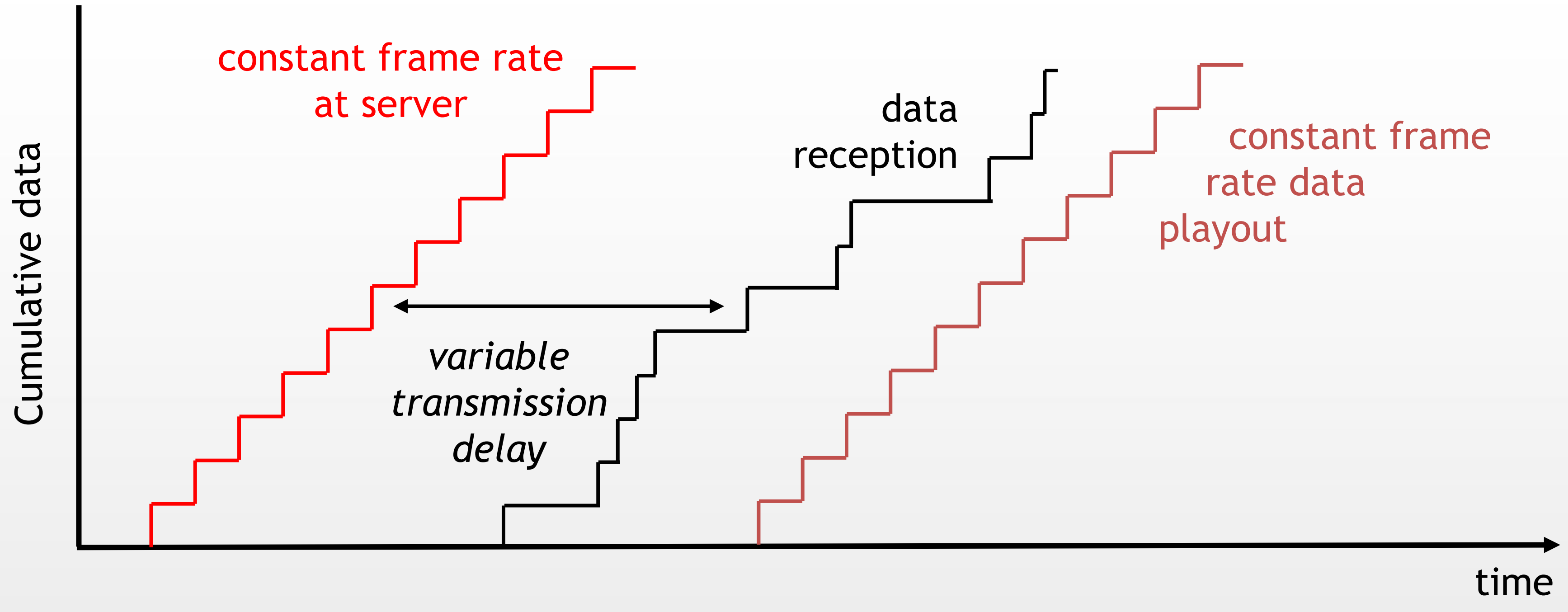
Khaled1: I saw that comeing you ugly bastard:
Attack state 13 ENEMY BIG WEAPON BEHIND you..Locked ON
to Enemi Yaw Difference: 0

Target Inside Box Speed Active.

How to Obtain a Uniform Inter-Arrival

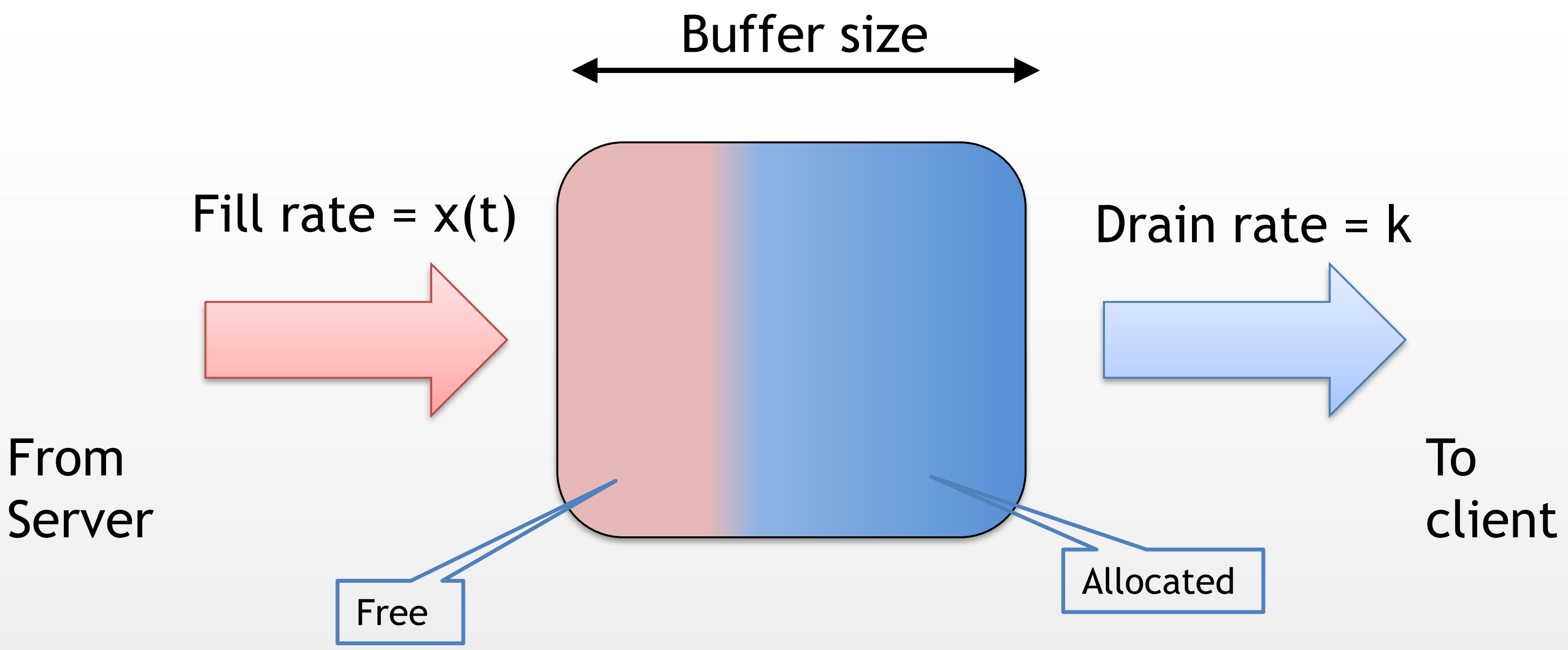
- Heaven: one frame update in each packet and one packet exactly every 1/60th of second
  - Too good to be true!
- Reality: this will NEVER happen
  - Internet does not follow a WFQ model (this has been formally proved)
- Jitter: the variation in latency as measured in the variability over time of packets latency across a network
  - Read: the instantaneous variation over the average packet inter-arrival time
  - We can measure it, but we cannot constrain it

Network and Variable Transmission Time



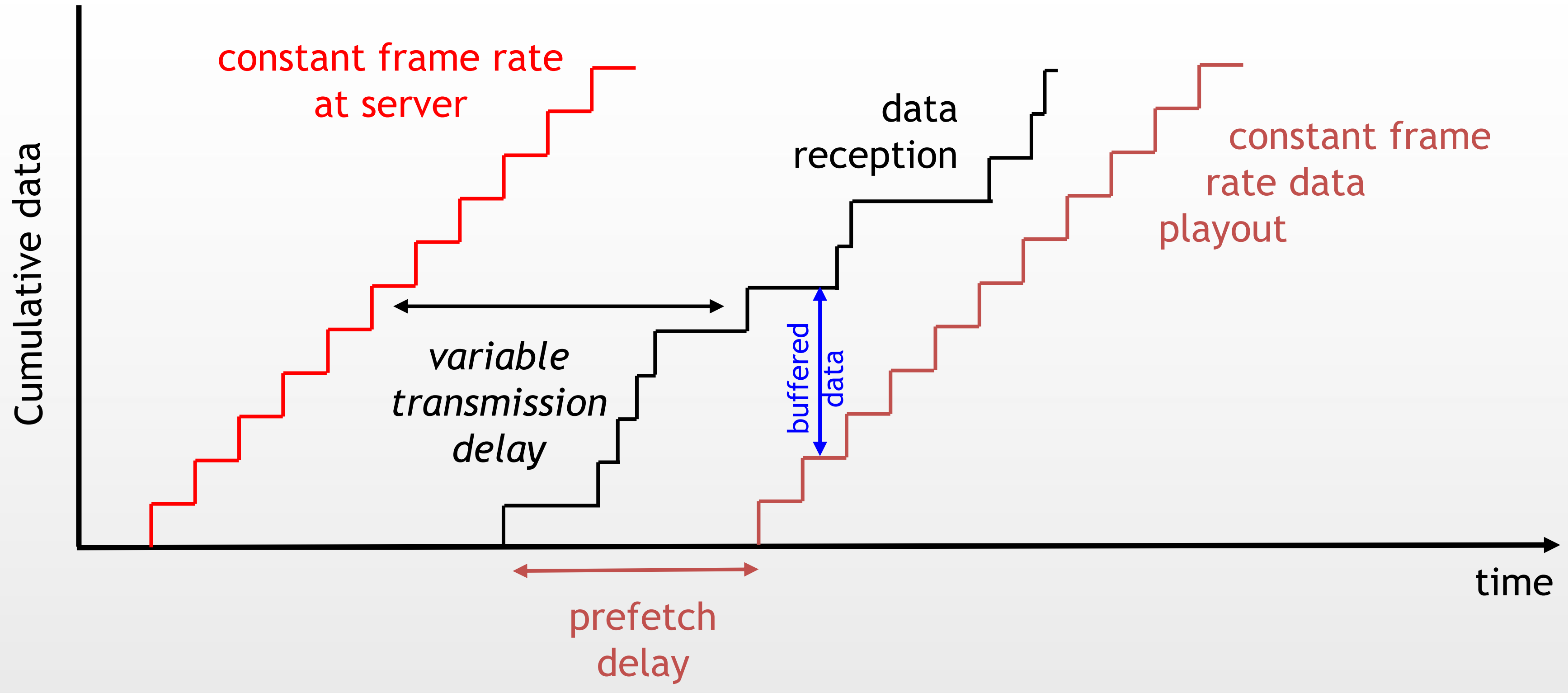
Buffering

The only way to ”convert” a variable frame rate into a constant one is to use a buffer



Unfortunately, the time spent in the buffer increases the delay even more

Network and Variable Transmission Time

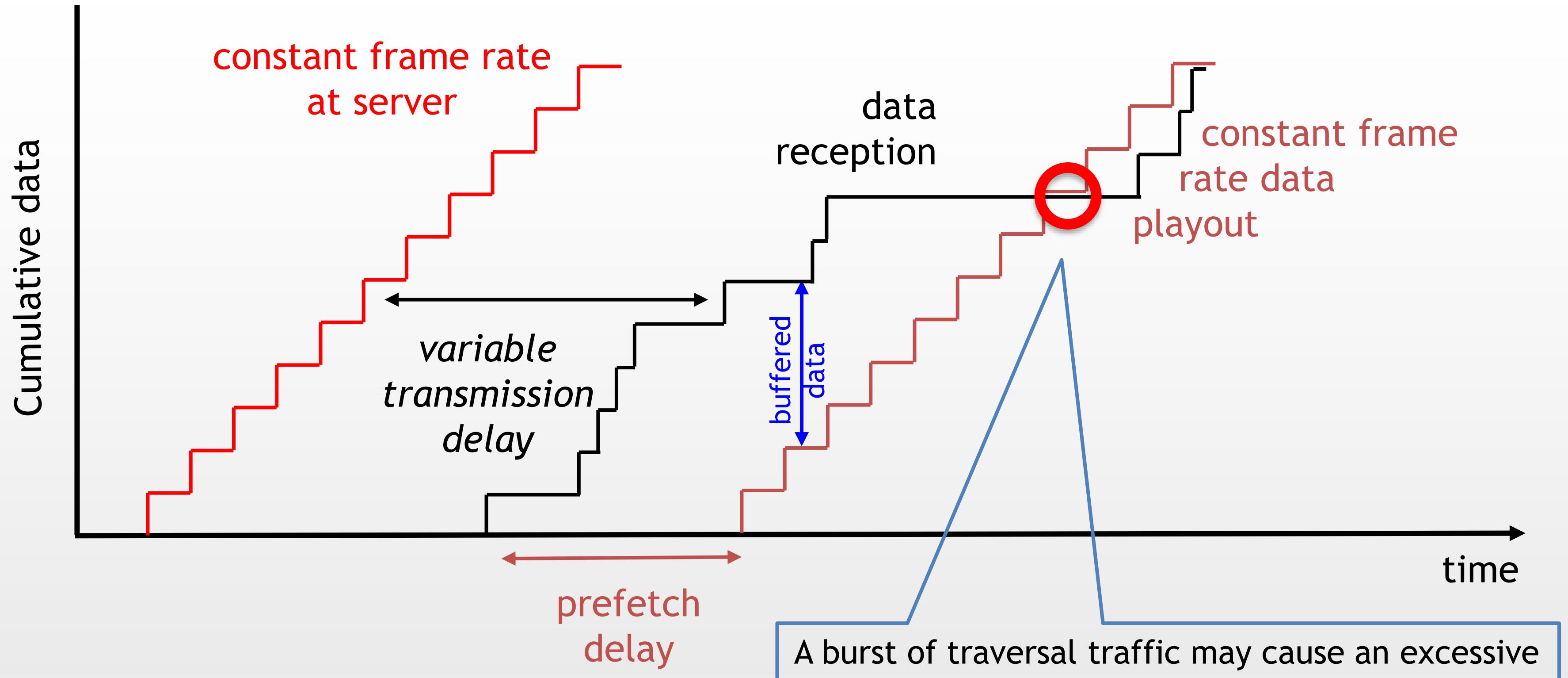

constant frame rate at server
data reception
constant frame rate data playout
Cumulative data
variable transmission delay
buffered data
time
prefetch delay

The Buffer Dilemma

- Buffer is too big
  - Large prefetch delay
  - Large increase of network LAG
  - Player experience will be poor

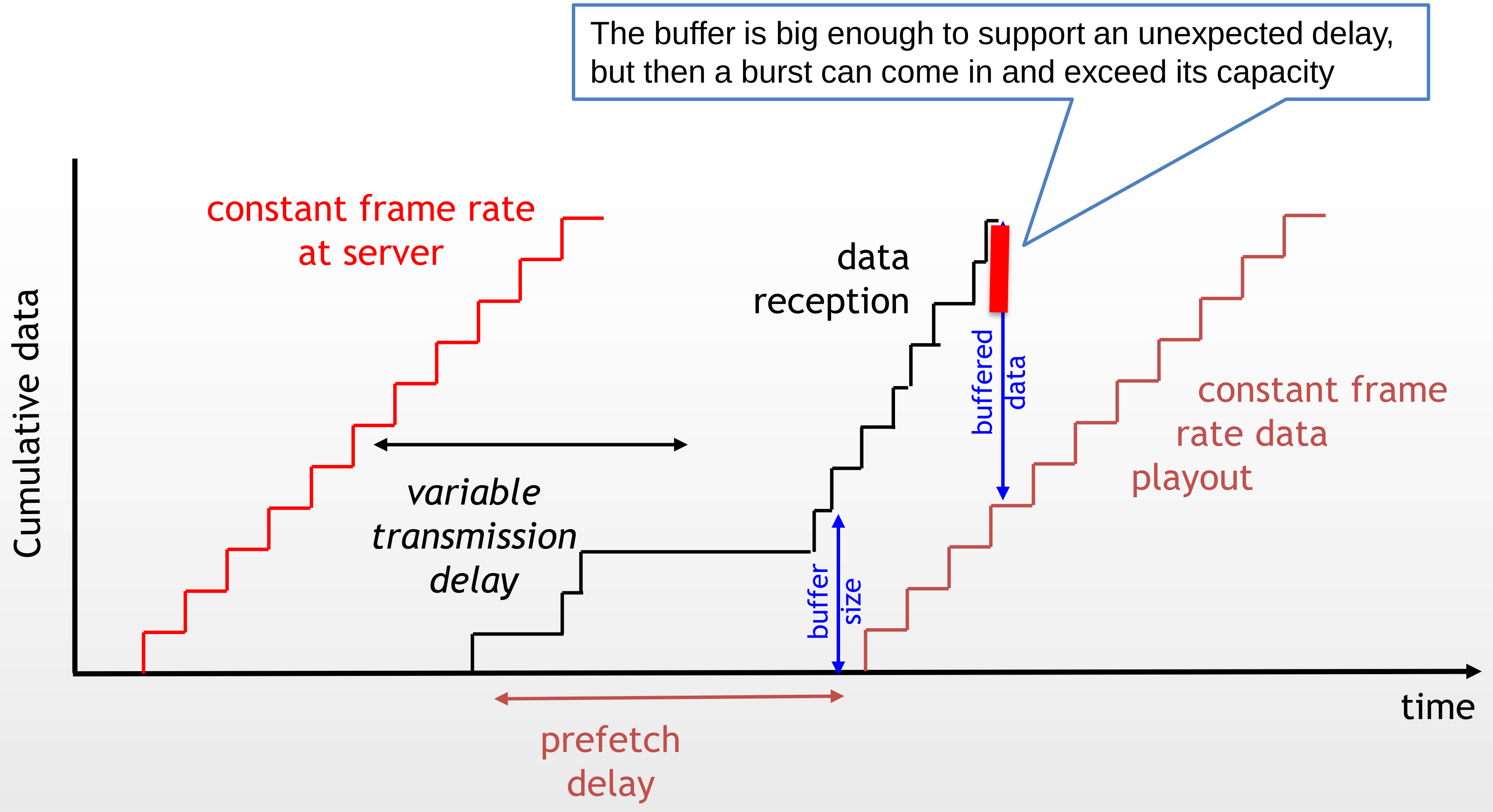
- Buffer is too small
  - Short prefetch delay
  - Small increase of network LAG
  - High Risk of buffer overrun or underrun
  - Player experience will be poor

Buffer Underrun


constant frame rate at server
data reception
constant frame rate data playout
variable transmission delay
buffered data
Cumulative data
time
prefetch delay

A burst of traversal traffic may cause an excessive delay and the buffer will be empty for a while, leaving the client with no data

Buffer Overrun


The buffer is big enough to support an unexpected delay, but then a burst can come in and exceed its capacity
constant frame rate at server
data reception
Cumulative data
variable transmission delay
buffered data
buffer size
constant frame rate data playout
time
prefetch delay

Making the Buffer “Just Right”

- We can use a somewhat large buffer while keeping a small prefetch
  - Less buffer overruns
  - Small increase of network LAG
  - Buffer underrun is still a problem
- To solve the underrun problem, we can raise an alarm whenever the buffer drops below a given threshold
  - This threshold should technically be the amount of data “along the way” over the network ... just like the optimal congestion window of TCP
  - We are assuming here we can measure correctly the network round trip time
  - TCP is not measuring anything, but uses a congestion avoidance algorithm which is not constant and so, useless for us
- What your game do when the alarm is raised is heavily dependent on your game
  - Read: it is up to you, or the networking middleware your engine is using

Packet Loss

- A lost packet is lost … forever
  live with it!

- If you needed that content so desperately …

1. Retransmit (as in TCP)
   ... and increase the delay due to timeouts

2. Be redundant (as in Forward Error Correction)
   … and increase network usage with extra traffic your user might not be willing to pay for

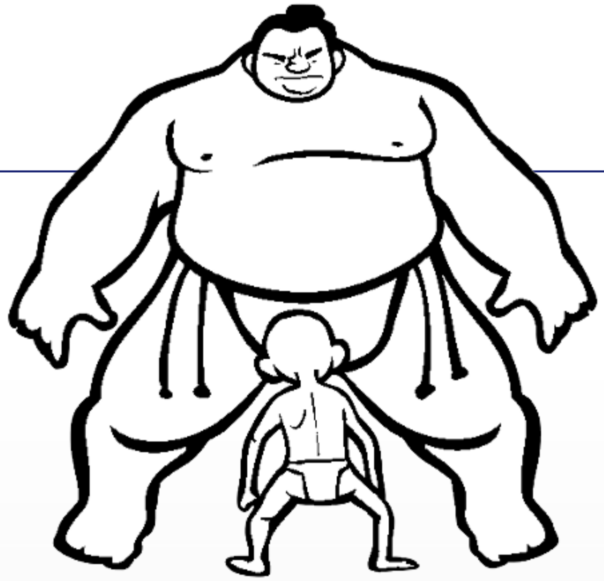
Summary: What Can we do Today ?

- For delay
  - LAG compensation (adding delay)
- For jitter
  - Add a buffer and take data out at a constant rate (adding more delay)

- How do they effect gameplay ?
  - This is still mostly unknown!

- The goal is to achieve fairness between players
  - Are we kidding? No player wants to be fair!

Fairness



- Be honest: no player wants to be fair
  - He/She/It just want to win!
- Fairness is a (game) provider problem
  - If the player feels to be on the “wrong side” of the deal he/she will just give up and drop the game
  - Providers want the game to be fair in order to keep users p[l]aying
- There is no such thing as “long term fairness”
  - We must prevent (or, at least, detect) unfairness in real-time

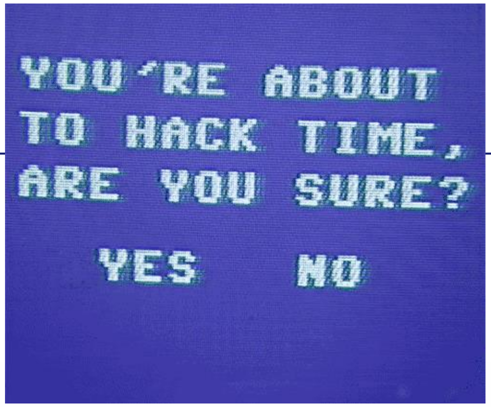
What Can We Really Do Today ?

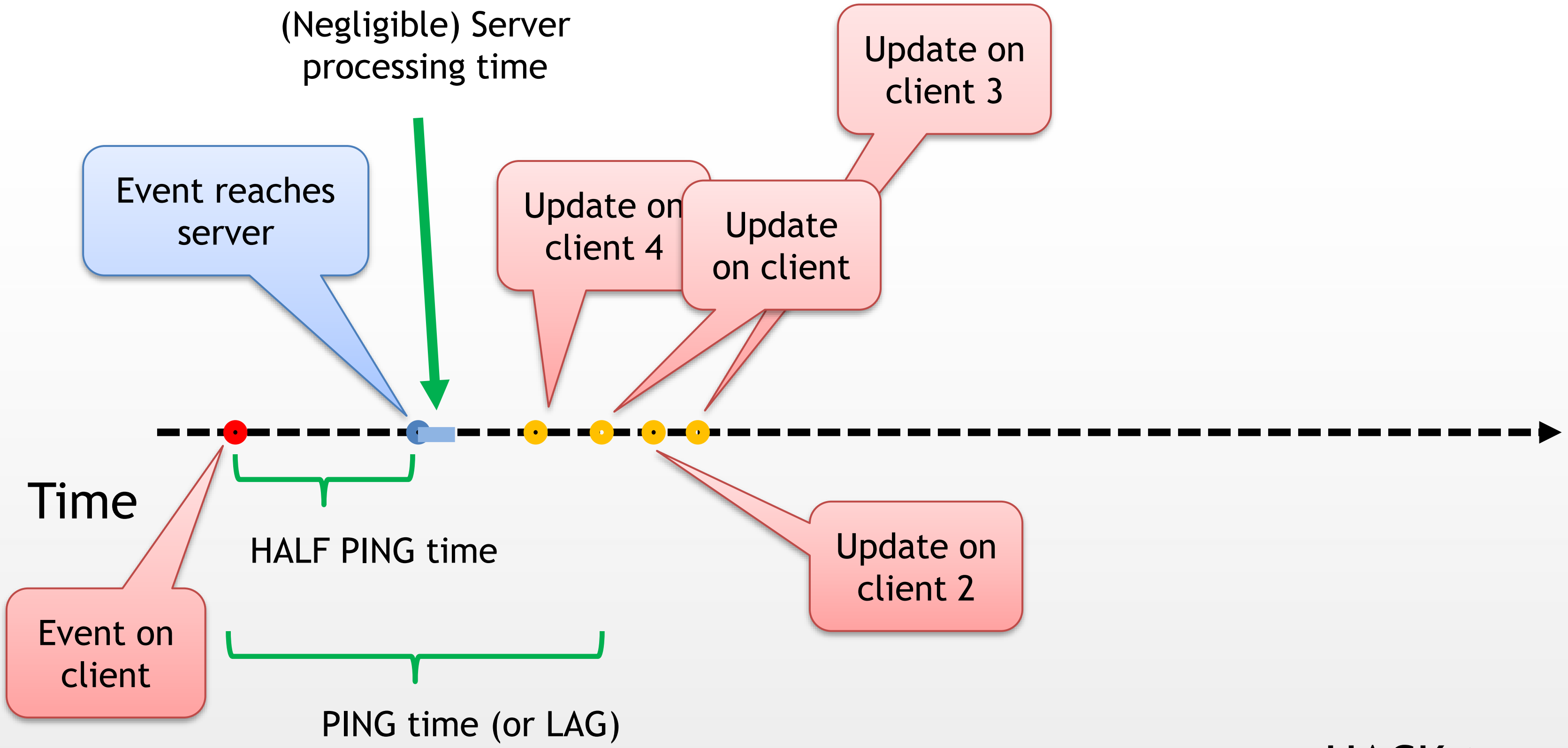
- Design and implement games AROUND network shortcomings
  - Implementation should target a technology/architecture first and set constraints
  - Game design should take into account network performances and design (constraint) game mechanics accordingly
  - Put server(s) as close as possible to users
    - Usually following the “bring home” approach of CDNs

- Players will adapt themselves

- Communities will cluster in servers based on observed LAG
  - Think about yourself when you want to play online

Understanding Time



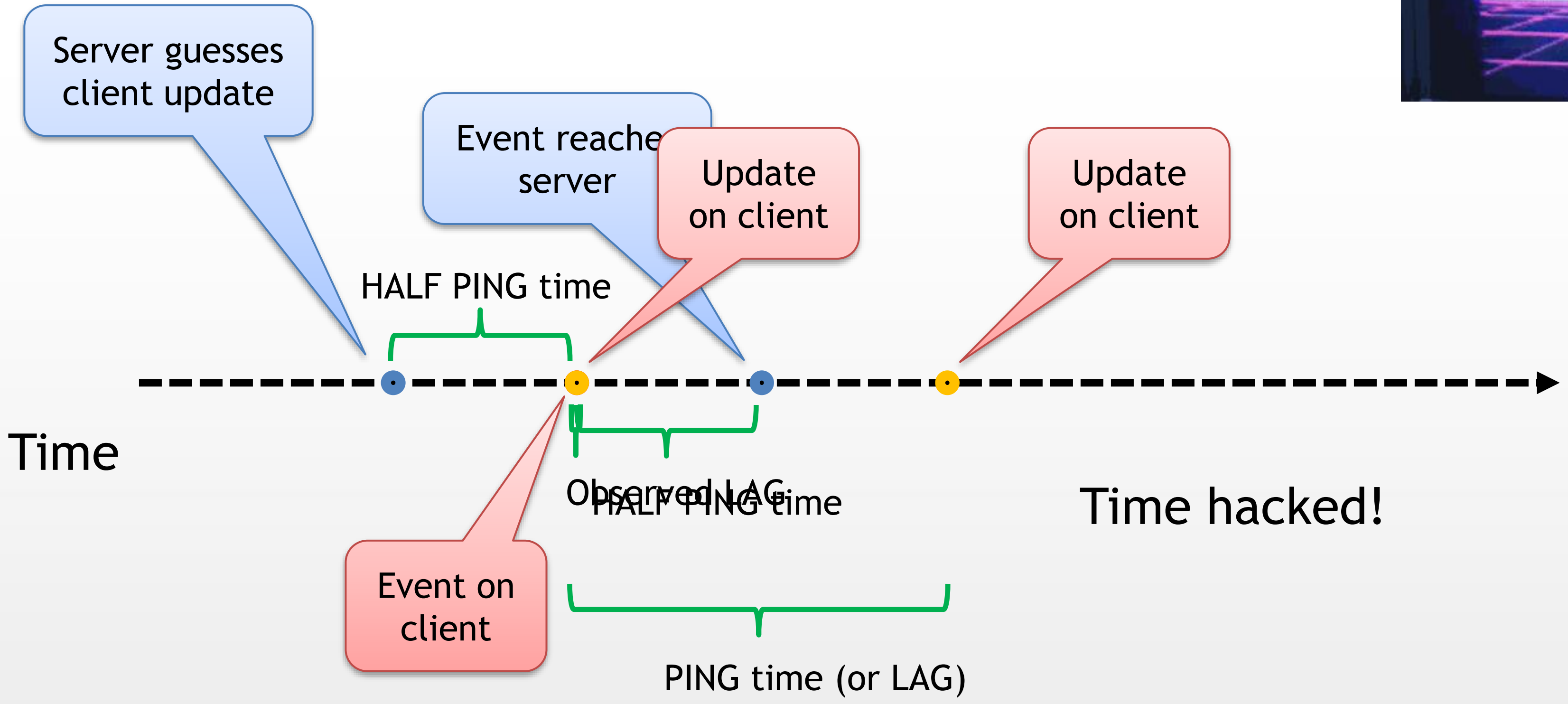


HACK target: reduce LAG

Hacking Time

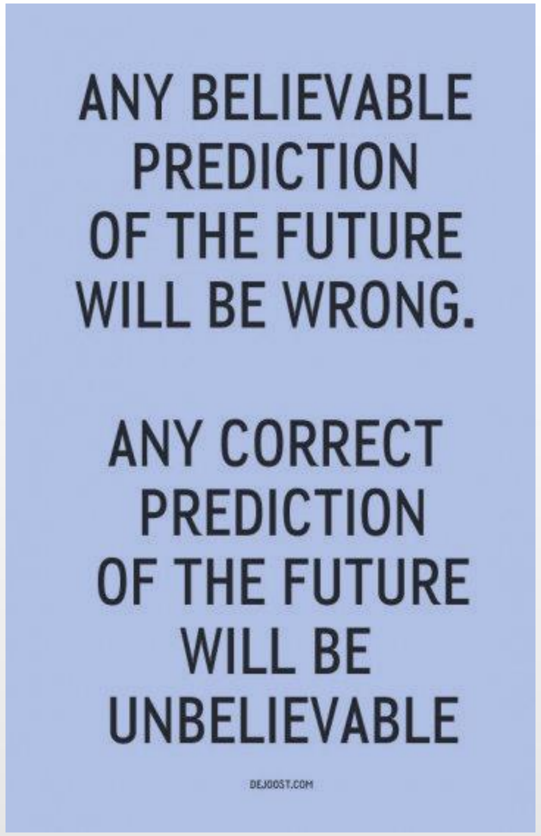
- Think about having a crystal ball to predict future
- What if you can “guess” the user input in advance?

Hacking Time



Prediction

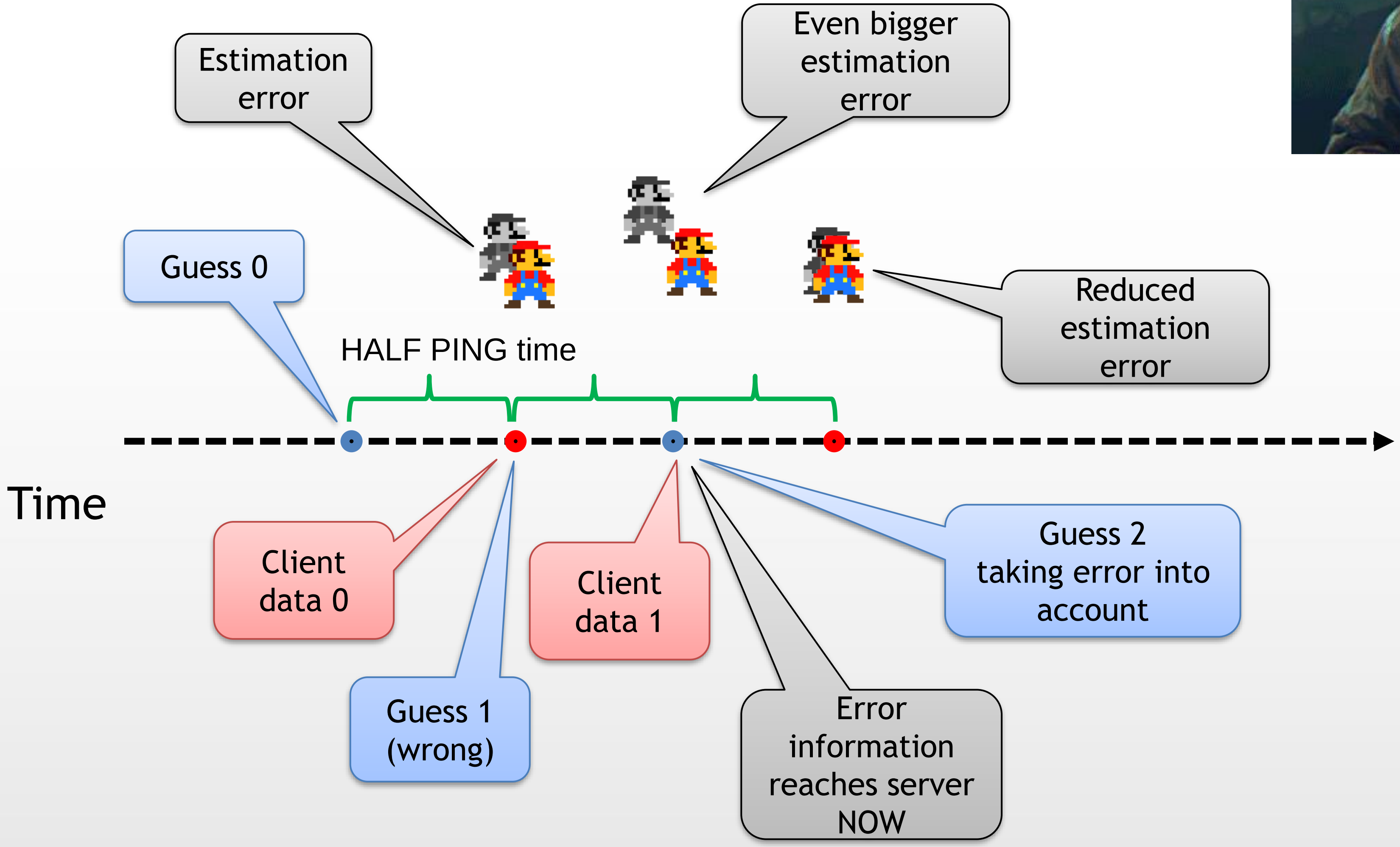
- Well, the problem now is how to make an accurate prediction
- No matter what we do ...
  players are unpredictable animals
- Then ... let's admit we can be wrong
  and see if there is a way to recover the error


ANY BELIEVABLE PREDICTION OF THE FUTURE WILL BE WRONG.

ANY CORRECT PREDICTION OF THE FUTURE WILL BE UNBELIEVABLE

DEJOOST.COM

Hacking Time



About Guessing

- Guessing means learning from the past inputs
  - NO, THIS IS NOT MACHINE LEARNING. Time is way too short for that
  - We interpolate over a time series
- Players have a limited number of options in a short period of time
  - So, the action space for guessing is not that large and errors may be limited
- No guessing is perfect
  - We will always have some error
  - But if the sampling time is short the error will be low

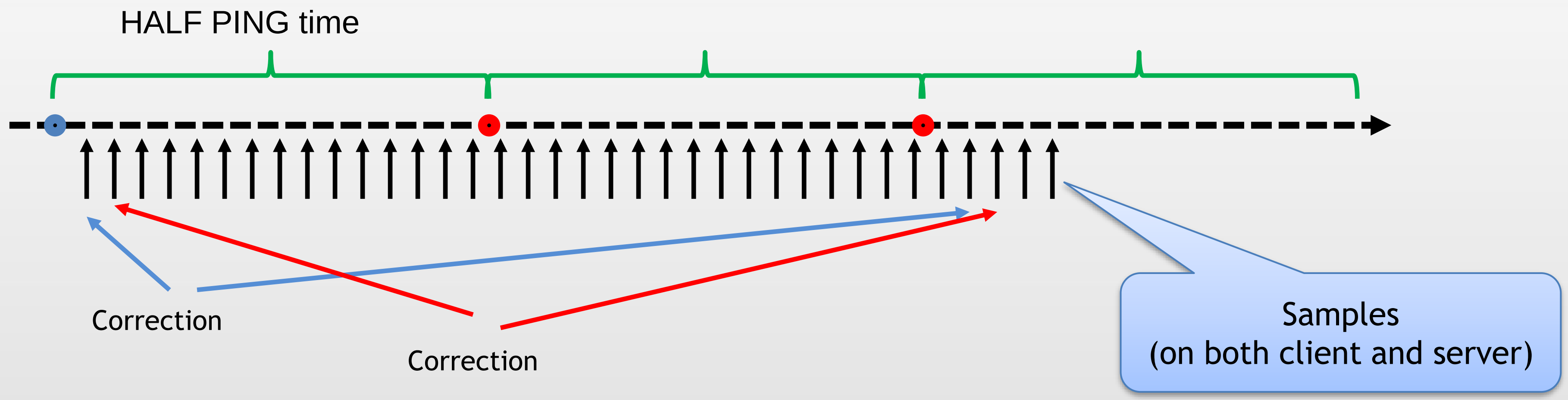
Let's take this as an opportunity!

- You **MUST** determine an admissible error starting from you game mechanics, or design mechanics based on the expected error

Reducing the Error

- The easy way: oversampling

- You send updates (guesses) and inputs much more often than the duration of the ping time
  - You will still get the error information after the ping time, but you can perform a correction every sample time.



Correction Strategies

- Warping
  - You put the player in the new position and that’s it
  - Acceptable only if error is bounded ... and small

- Lerping
  - You gradually correct the player to have it “swerve” in the right position
  - Takes longer (more than one sample) and will lose precision
  - More natural, but also more computational-intensive

Future Will be Troublesome

- Bigger and bigger content
  - Will it fit on my pipeline?
- Players want it streaming (and now!)
  - How much should I wait to start playing?
- Players want to put their own contents
  - Video (as in YouTube)
  - Audio (as in mySpace)
  - Environment (as in Second Life)
    - And YOU will be legally responsible for the content
- How long can we keep saying that "*bounding the delay*" is enough?

References

- Networking and Online Games
  ISBN 0-470-01857-7
  by Armitage & all.
  § 5 and § 6

- Integrated Services in the Internet Architecture: an Overview
  RFC1633, online: https://tools.ietf.org/html/rfc1633
  by Baden & all.
  Up to § 2.1 (included)

- An Architecture for Differentiated Services
  RFC2475, online: https://tools.ietf.org/html/rfc1633
  by Blake & all.
  § 1.1, 2.1, and 2.2

- Source Multiplayer Networking
  online: https://developer.valvesoftware.com/wiki/Source_Multiplayer_Networking
  by Valve