

TCP and Computer Games

Presented by Garena Online Pte. Ltd.

Garena



- Revenue more than tripled every year since 2009
- More than 500 employees in branches
 - Singapore Malaysia
 - Taiwan Thailand
 - Philippine Vietnam
 - Indonesia Kazakhstan
 - ...
- It began with a 10-person team and one product for playing LAN games over Internet.

Still Remember LAN Games?



Warcraft 3



Dota

Why play LAN game on Internet is difficult?

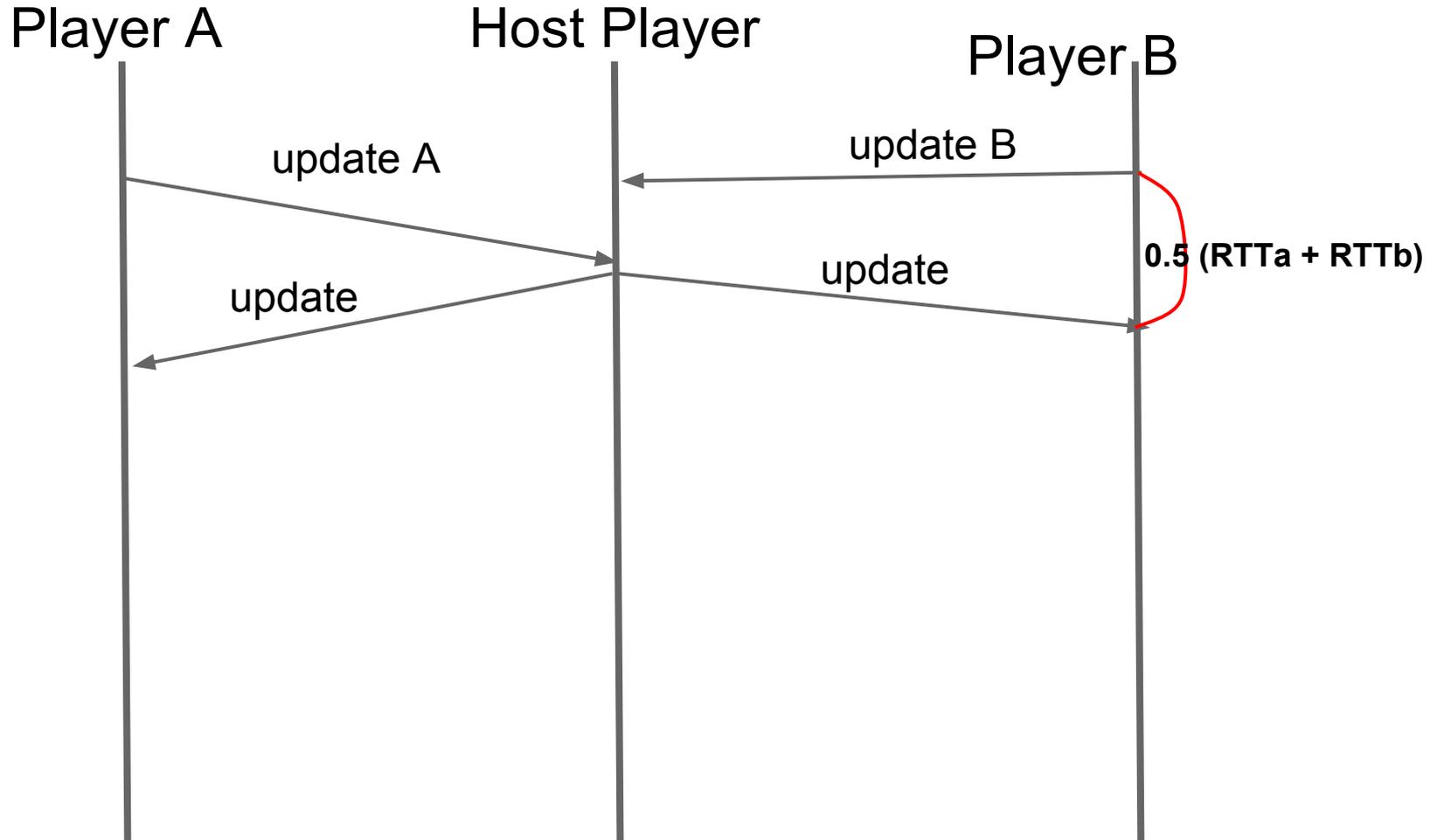
LAN

- low latency
 - < 1ms
- low lossrate
 - almost 0

Internet

- latency can be high
 - 50ms - 1s
- lossrate can be high
 - sometimes 1% - 5%

Responsiveness is determined by the player having largest RTT



Suppose the tolerable latency is 0.6-0.8s for DoTA, theoretically we can allow users with ping < 0.8s to join the game.

Garena LAN game Lobby will list the ping value of users

Local Player 1	100ms
Local Player 2	200ms
Local Player 3	300ms
Oversea Player	800ms

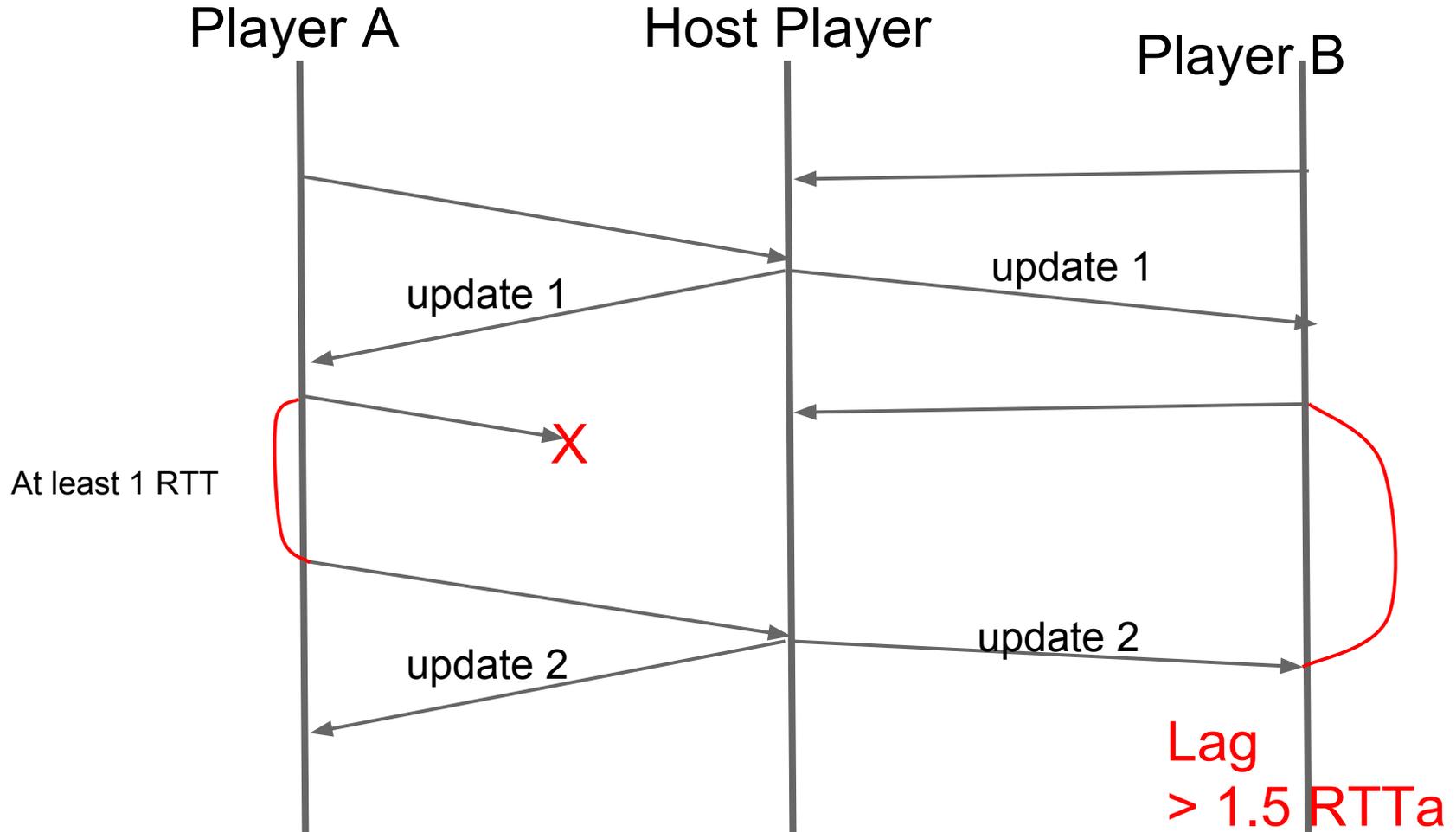
But,

- TCP is commonly used for status updating in LAN games.
- High latency usually correlates to high lossrate.
 - Why?

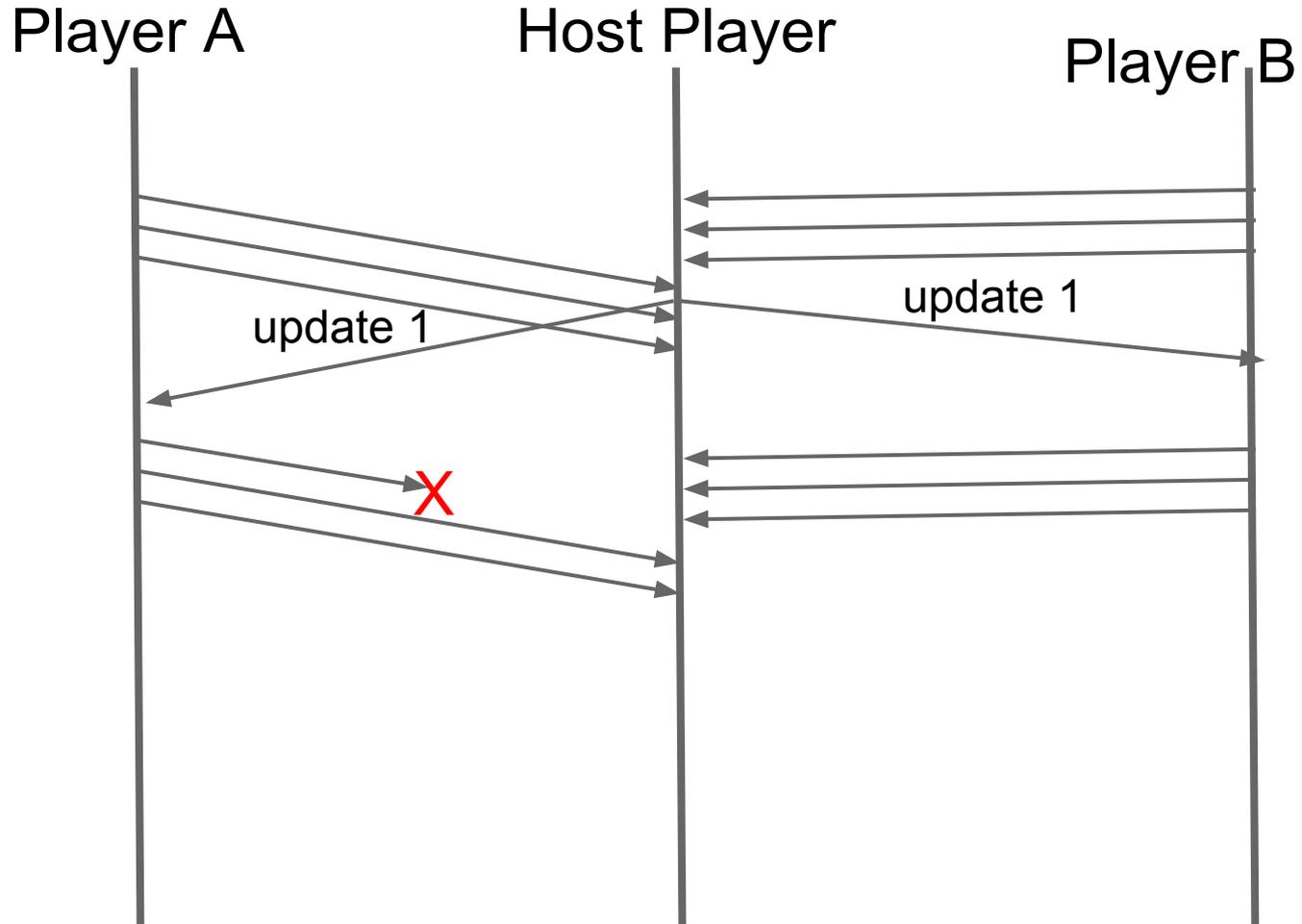
But,

- TCP is commonly used for status updating in LAN games.
- High latency usually correlates to high lossrate.
 - Why?
 - high latency
 - > high queueing delay
 - > long queues in the routers
 - > higher chance for packets be dropped by the router.

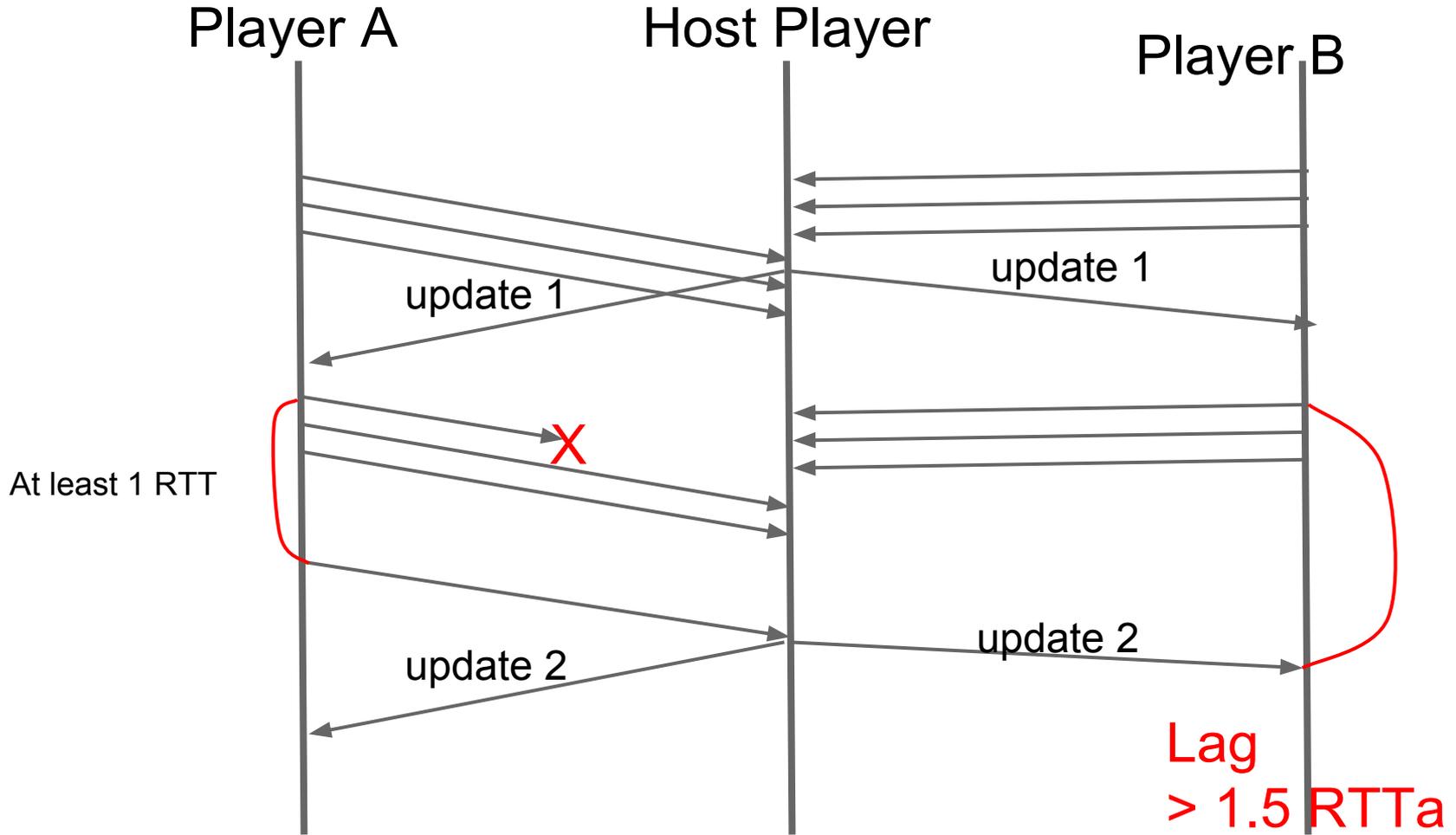
How about if a packet is lost, ...



Since the packet is usually small, how about send it multiple times, ...

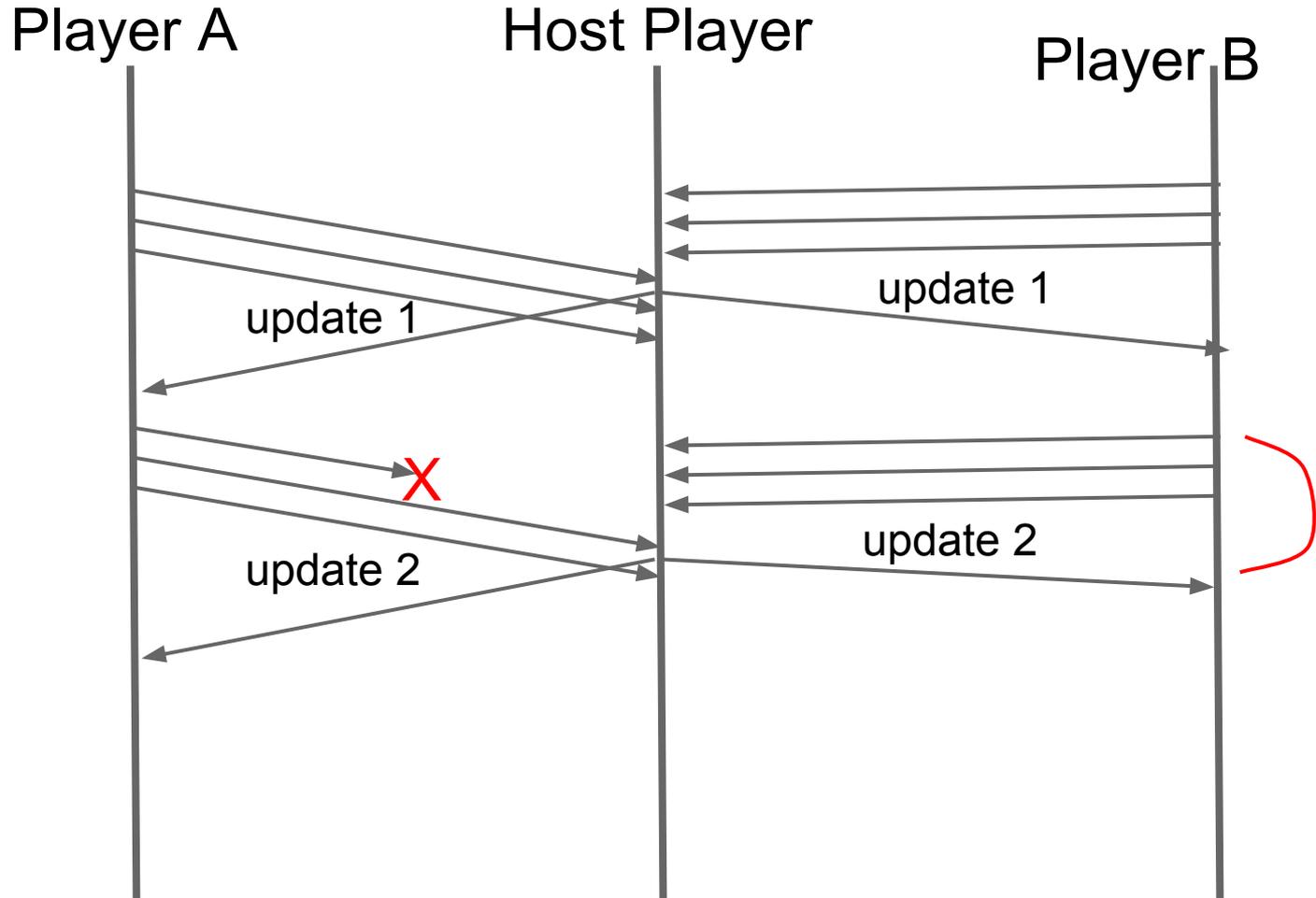


It does not help because TCP will not deliver the late packets until the one lost is retransmitted, ...



Solution?

Use UDP instead

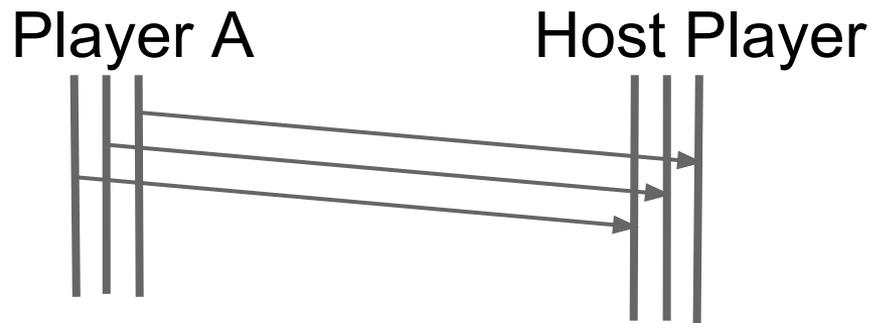


- FEC instead of retransmission to reduce latency
- UDP is a better choice when FEC is needed.
- Question:
If only TCP is possible, can we do FEC?

- FEC instead of retransmission to reduce latency
- UDP is a better choice when FEC is needed.
- Question:

If only TCP is possible, can we do FEC?

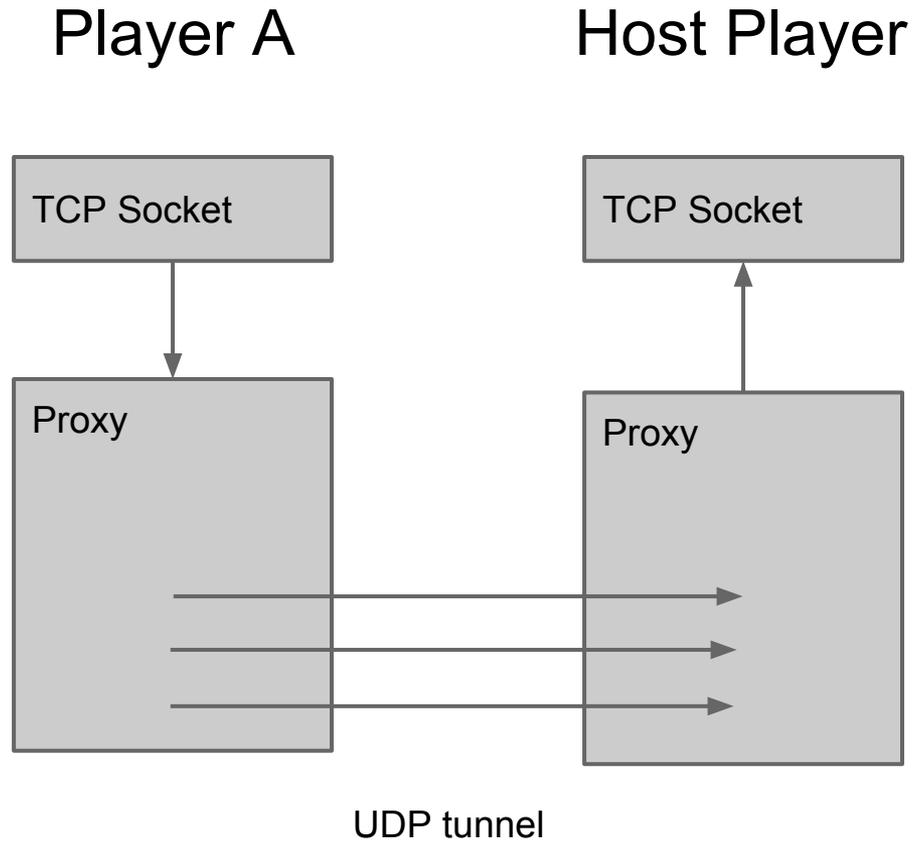
Yes, we can use multiple connections



Create 3 connections and send the same packet in all 3 links.
There are, however, still unnecessary retransmissions.

How about the existing games? We cannot change their protocols.

UDP tunnel



Rule of Thumb 1:

Try to avoid using TCP if low latency is required.

TCP is still widely used in games where latency is insensitive.

- Login
- Chatting
- Online shopping/trading

Reason

- Simple to use
 - it handles packet losses for us.
 - it handles packet duplication for us.
 - it handles packet disorders for us.

So with TCP, developers do not need to handle packet loss and packet duplication anymore.



So with TCP, developers do not need to handle packet loss and packet duplication anymore.

Unfortunately, it is wrong.

‘So with TCP, developers do not need to handle packet loss and packet duplication anymore.’

It’s only true when a TCP connection is never broken in the middle of transmission.

But,

1. for a server running 24/7, unexpected TCP disconnection is inevitable.
2. for a game with tens of thousands players online, one or two of them get unexpected TCP disconnection is almost certain.
3. smartphones, especially when they are mobile, suffer from a lot of disconnections.

In many scenarios, we need to do auto reconnection so players will not notice it.

How about the data sent just before the disconnection?

Can we precisely know which part of data has been sent, which not?

Can we precisely know which part of data has been sent, which not?

No, so we have to introduce application layer acks if no data loss is allowed.

- actually on Linux, there's a way to know how many bytes have been acked
- getsockoptinfo with TCP_INFO
- it is not available in Windows, and even on Linux it's quite troublesome

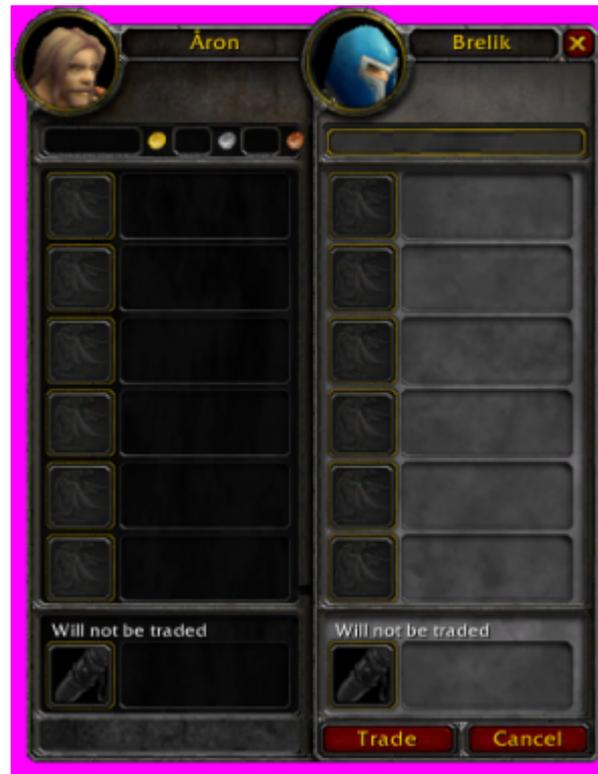
We are all set if we always resend the un-acked data, right?

No, We can only be sure which packets have been sent, but not sure about the loss, because



With application layer Acks, duplicated data have to be handled carefully.

Online trading



Duplicated packets may cause unexpected double trades (or worse, double payments).



- transfer 2 gold coins from A to B
- transfer 2 gold coins from A to B

Solution?

Solution: Idempotence

A system is idempotent when an operation can be done multiple times without changing the result.

How to be Idempotent?

Assign unique ID to each operation, and ignore the one has been processed.

- transfer 2 gold coins from A to B
- transfer 2 gold coins from A to B

How to be Idempotent?

Assign unique ID to each operation, and ignore the one has been processed.

- TxnId 1abc1aef, transfer 2 gold coins from A to B
- TxnId 1abc1aef, transfer 2 gold coins from A to B

How to be Idempotent?

Assign unique ID to each operation, and ignore the one has been processed.

Be careful if there are multiple processing threads.

- always process the operation with the same ID in the same thread.
- or use lock properly.

Rule of Thumb 2:

Prepare handling data loss or duplicate data even when you are using TCP.

Wrap up

1. Prefer (UDP + FEC) to TCP for latency sensitive communications.
2. Don't forget packet loss and duplication even only TCP is used.

Finally

Small innovation may lead to big company.

