Fairness in Multi-Player Online Games on Deadline-Based Networks

Anh Le, Yanni Ellen Liu
University of Manitoba (2006)

CCNC 2007
Las Vegas, Nevada
Outline

1. Introduction
2. Unfair Scenarios in MOGs
3. Fairness Metrics
4. Fairness Strategies
5. Performance Evaluation
6. Conclusion
Deadline-Based Networks

- Real-time applications: online auction, Internet telephony, interactive online game, etc.

- Delays: transmission, propagation, processing, and *queuing delays*
Real-time applications: online auction, Internet telephony, interactive online game, etc.

Delays: transmission, propagation, processing, and *queuing delays*

Deadline-based networks:
- Each Application Data Unit (ADU) is associated with a deadline
- ADU deadlines are mapped to packet deadlines and carried in packets
- Deadline-based scheduling is employed in routers
- Reduce *queuing delays*
Multi-player online games: World of Warcraft, Counter Strike, etc.
Stringent requirement on delay performance
Multi-player online games: World of Warcraft, Counter Strike, etc.

Stringent requirement on delay performance

Fairness:
- Equal opportunity to win the game
- Unfairness caused by differences among end-to-end delays
Multi-player online games: World of Warcraft, Counter Strike, etc.

Stringent requirement on delay performance

Fairness:
- Equal opportunity to win the game
- Unfairness caused by differences among end-to-end delays

Strategies: network-based and delay-compensation
Unfair Scenario 1 - Client to Server

Server receives 1 killed A and assumes 2 killed A first.

Server receives 2 killed A and credits 2 for the kill.

Client 2 kills object A

Client 1 kills object A

Time

Client 1 to Server delay

Client 2 to Server delay

Client 1

Client 2
Unfair Scenario 1 - Client to Server

Server receives 1 killed A and assumes 2 killed A first.

Server receives 2 killed A and credits 2 for the kill.

Client 2 kills object A

Client 1 kills object A

Client 1 to Server delay

Client 2 to Server delay

Client 1

Client 2

Unfair to client 1!
Unfair Scenario 2 - Server to Client

Client 1 receives the message but already dead

Client 2 receives the message shoots and kills Client 1

Server sends message to both clients saying that they can see each other

Server to Client 1 delay

Server to Client 2 delay
Unfair Scenario 2 - Server to Client

Client 1 receives the message but already dead

Client 2 receives the message shoots and kills Client 1

Server sends message to both clients saying that they can see each other

Server to Client 1 delay

Server to Client 2 delay

Unfair to client 1!
Definition of Fairness

A game session is said to be fair if:

- The *average* one-way delays of the packets sent from each *client to the server* are the same
- The *average* one-way delays of the packets sent from *the server to each client* are the same
- $n$: number of clients
- $(d_i)_k (i = 1 \ldots n)$: end-to-end delay of packet $k$ from client $i$ to the server
- $m_i$: number of packets sent from client $i$ to the server
- $\theta_i (i = 1 \ldots n)$: average end-to-end delay of all the packets sent from client $i$ to the server

\[
\theta_i = \frac{\sum_{k=1}^{m_i} (d_i)_k}{m_i}
\]
\( \epsilon \): average end-to-end delay of all packets sent from all the clients to the server

\[
\epsilon = \frac{\sum_{i=1}^{n} \theta_i}{n}
\]

\( \delta_i \) \((i = 1 \ldots n)\): absolute value of the difference between \( \theta_i \) and \( \epsilon \)

\[
\delta_i = |\theta_i - \epsilon|
\]

Fairness index 1, denoted by \( F_1 \)

\[
F_1 = \sum_{i=1}^{n} \delta_i
\]
In the interval $I_{j+1}$, extra delays are added to the packets sent from clients who have the average delay $\theta$'s lower than the total average delay $\epsilon$ in the interval $I_j$.

Amount of delay added is: $\epsilon - \theta_i (i = 1 \ldots n)$
In the interval $I_{j+1}$, deadlines of the packets sent from clients, who have the average delay $\theta$'s larger than the total average delay $\epsilon$ in the interval $I_j$, are adjusted downwards.

New end to end deadline must be larger than (total transmission delay + total propagation delay)
Strategies Illustration

Average Client-to-Server
End-to-End delay

Client 1  Client 2  Client 3
In the next interval:

- **Delay-Compensation**: Add extra delays
In the next interval:

1. Delay-Compensation: Add extra delays
2. Network-Based: Use more urgent deadlines
Performance Model

- Discrete event simulation model developed in Java
- Network model: a scaled down version of Abilene backbone network (http://abilene.internet2.edu/)
- Traffic model: foreground first person shooter (FPS) game, background FPS game, music, movie
Network Model

1. Seattle
2. Denver
3. Kansas
4. ...
Effect of Network Load on F1

![Bar chart showing the effect of network load on F1 value. The chart compares F1 values for different bottleneck utilizations (95.00%, 90.00%, 85.00%) with and without strategies.]
Effect of Deadline Assignment Scheme on F1

- Scheme i
  - No Strategies
  - Network Based Strategy
  - Delay Compensation Strategy
  - Both Strategies

- Scheme ii
  - No Strategies
  - Network Based Strategy
  - Delay Compensation Strategy
  - Both Strategies

F1 Value (ms)

Deadline Assignment Scheme
Conclusion

- Fairness in MOGs is defined
- Fairness metrics are introduced

Both Delay-Compensation and Network-Based strategies can improve fairness significantly

- Delay-Compensation can be used on current network
- Network-Based requires Deadline-Based network