Networking, Traffic Jams, and Schrödinger's Cat

Shawn Hargreaves
Software Development Engineer
XNA Community Game Platform
Microsoft
XNA Framework Networking

- I spoke about networking at Gamefest 2007
- What the framework does for you
  - Finding and joining sessions
  - Synchronizing the list of players
  - Lobby transitions
  - Reliable UDP protocol
  - Voice ‘just works’ ™
Confession

- My 2007 networking demo made some simplifying assumptions
  - Computer programs are deterministic
  - Bandwidth is infinite
  - Speed of light is fast enough to be irrelevant
The Inconvenient Truth

- Networked programs are only deterministic if you ignore time
- Bandwidth is far from infinite
- The speed of light is actually pretty slow!
The Five Stages of Networking

- Denial
- Anger
- Bargaining
- Depression
- Acceptance
DENIAL
The Ostrich Technique

- My game is small and simple
- 8 players
- 30 fps
- Each frame I send
  - Position : Vector3
  - Velocity : Vector3
  - IsFiring : bool

\[
\begin{align*}
7 \times 30 \times (12 + 12 + 1) &= 5.1 \text{ k}
\end{align*}
\]
Packet Header Bandwidth

- IP header: 20 bytes
- UDP header: 8 bytes
- LIVE: 16 bytes
- XNA Framework: ~7 bytes
- = ~51 bytes
## Game Data + Headers

- 8 players
- 30 fps
- Each frame I send:
  - Position : Vector3
  - Velocity : Vector3
  - IsFiring : bool
  - Packet header

\[
\text{7 x} \quad (12 + 12 + 1 + 51) \quad = 15.6 \text{ k}
\]

- Total size: 15.6 k
Voice Bandwidth

- Voice data is approximately 500 bytes per second.
- By default, all players can talk to everyone.
Game Data + Headers + Voice

- 8 players
- 30 fps
- Each frame I send
  - Position : Vector3
  - Velocity : Vector3
  - IsFiring : bool
  - Packet header
- Voice data

Recommended limit is 8 kilobytes per second

\[7 \times (30 \times (12 + 12 + 1 + 51) + 500) = 19 \text{ k}\]
ANGER
WTF?

- Why so little bandwidth?
- My service provider claims way more
- Other apps report way more
Games Are Unforgiving

- The Internet is highly variable
- Service providers advertise their best case
- Web browsers care about the average
  - Occasional spikes are no problem
- Games are limited by the worst case
  - A 30 second glitch is enough to ruin gameplay
BARGAINING
My Morning Commute
# Xbox LIVE Console Bandwidth

% of consoles with this bandwidth or less | **Downstream bandwidth** (*) | **Upstream bandwidth** (*)
--- | --- | ---
0.1% | 2.5 | 3
0.5% | 5 | 5.5
1% | 8 | 8
2% | 9 | 10
5% | 13 | 15
10% | 18 | 22
30% | 25 | 32
50% | 42 (median) | 44 (median)
70% | 113 | 115
90% | 648 | 693

Thanks to Bungie Studios for data collected during Halo 3 beta, 2007

* Bandwidth in kilobytes per second
Latency

- Speed of light = 186,282 miles per second
- Nothing can travel faster than this
- Seattle to L.A.
  - 960 miles = 5 milliseconds
- Seattle to England
  - 4,799 miles = 26 milliseconds
- Speed of light in fiber or copper slows to 60%
- Each router adds 5 to 50 ms
- DSL or cable modem adds 10 ms
# Xbox LIVE Console Latency

<table>
<thead>
<tr>
<th>% of consoles with this latency or less</th>
<th>Round-trip latency between consoles (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10%</td>
<td>32</td>
</tr>
<tr>
<td>30%</td>
<td>57</td>
</tr>
<tr>
<td>50%</td>
<td>84 (median)</td>
</tr>
<tr>
<td>70%</td>
<td>130</td>
</tr>
<tr>
<td>90%</td>
<td>250</td>
</tr>
<tr>
<td>95%</td>
<td>320</td>
</tr>
<tr>
<td>97%</td>
<td>380</td>
</tr>
<tr>
<td>98%</td>
<td>430</td>
</tr>
<tr>
<td>99%</td>
<td>540</td>
</tr>
<tr>
<td>99.9%</td>
<td>770</td>
</tr>
</tbody>
</table>

*Thanks to Bungie Studios for data collected during Halo 3 beta, 2007*
DEPRESSION
Packet Loss

- **SendDataOptions.None**
  - Packets may never arrive, or may get jumbled

- **SendDataOptions.InOrder**
  - Cheap

- **SendDataOptions.Reliable**
  - Costs bandwidth

- **Reliable + InOrder**
  - Costs bandwidth and latency
ACCEPTANCE
Know Your Limits

- **Bandwidth (kilobytes per second)**
  - Typical: 12 to 250
  - Worst case: 8

- **Latency (one-way)**
  - Typical: 25 to 250 ms
  - Worst case: 270 ms

- **Packet loss**
  - Typical: 2%
  - Worst case: 10%
Live Within Your Means

NetworkSession properties
- BytesPerSecondSent
- BytesPerSecondReceived
- SimulatedLatency
- SimulatedPacketLoss
Send Fewer Packets

- Trade bandwidth for latency
- Send data less often
  - Typically 10 to 20 times per second
- Prefer few big packets to many small ones
- Automatic packet merging
  - Multiple sends before NetworkSession.Update
  - Combines into a single wire packet
Send Smaller Packets

- Generalized compression algorithms are not much use
- Send smaller data types
  - int -> byte
  - Bitfields
  - Matrix -> Quaternion + Vector3
  - Spawn position -> spawn point index
  - Avoid strings!
Send Less Voice Data

- LocalNetworkGamer.EnableSendVoice
- Only talk to players on your team
- Only talk to people near you in the world
- But avoid changing this too often
Know What Matters

- Some things matter a lot
  - Am I dead?
  - Who picked up the Pan Galactic Gargle Blaster?
  - Who won?

- Some things only matter a little bit
  - Where am I?
  - What direction am I moving?

- Some things don’t matter at all
  - Which way did the dust particle bounce?
Distribute Your Workload

- PC games are usually client/server
  - Reduces cheating
  - Concentrates bandwidth load on one machine
- Xbox LIVE is more secure
- Peer-to-peer is a viable option
  - Avoids round-trip latency
  - Harder to maintain consistency
Hybrid Topologies

- Server makes important decisions
- Peers control most object movement
- Multiple specialized authorities
  - One machine controls powerups
  - Another tracks the score
Living With Latency

- Embrace quantum uncertainty
  - You can never know where an object is
  - Only where it used to be
  - ... and how fast it was moving
- The current state is a probability field
  - *Is the cat alive or dead?*
- Each player has their own parallel universe
- Our goal is to keep the universes similar
  - *No red pill*
Prediction

Prediction Relativity

- Remember the parallel universes
- Send information relative to the recipient
  - ‘Fired toward position \((x, y, z)\)’
- What if players are in different places?
  - ‘Shot at Shawn, missed 10° to the left’
- Robust even if player positions differ
Prediction Paradox

- Larger packets = less bandwidth
- Position + velocity + controller input
- More data makes prediction work better
- Allows lower packet send rate
- Fewer packet headers