Provisioning On-line Games: A Traffic Analysis of a Busy Counter-Strike Server

Wu-chang Feng, Francis Chang, Wu-chi Feng, Jonathan Walpole

OGI SCHOOL OF SCIENCE & ENGINEERING
OREGON HEALTH & SCIENCE UNIVERSITY

OpenOffice.org

Goal

- Understand the resource requirements of a popular on-line FPS (first-person shooter) game

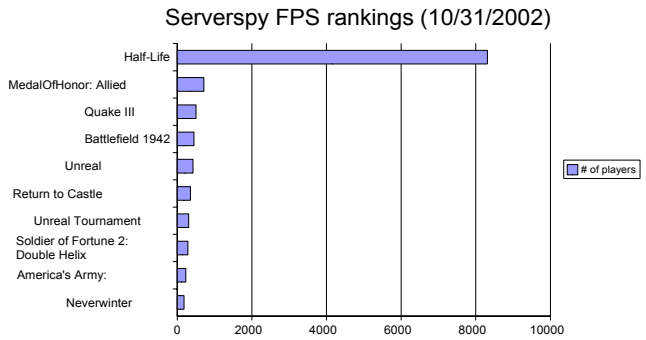
Why games?

- Rapidly increasing in popularity
  - Forrester Research: 18 million on-line in 2001
  - Consoles on-line
    - Playstation 2 on-line (9/2002)
    - Xbox Live (12/2002)
  - Cell phones
    - Nokia Doom port (yesterday)

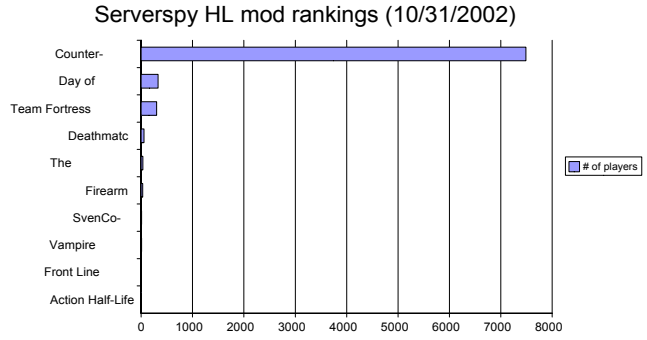
Why FPS?

- Gaming traffic dominated by first-person shooter genre (FPS) [McCreary00]

Why CS?



Why CS?



Networked FPS lineage

Doom

↓

Doom II → Quake

+ QuakeWorld variants
+ Team Fortress
+ Capture the Flag

↓

Quake II

+ Soldier of Fortune
+ Heretic II

↙ ↘

Quake III Arena

+ Medal of Honor Allied Assault
+ Return to Castle Wolfenstein
+ Soldier of Fortune 2
+ Jedi Knight II

↓

Doom III

Half-Life

+ Counter-Strike
+ Day of Defeat
+ Urban Terror
+ Team Fortress Classic
+ Team Fortress 2

Unreal

↓

Unreal Tournament

↓

Unreal Tournament 2003

+ America's Army: Operations

8 of top 10 games derived from one of two lineages

Counter-Strike

About the game...

- Half-Life modification
- Two squads of players competing in rounds lasting several minutes
- Rounds played on maps that are rotated over time
- Each server supports up to 32 players

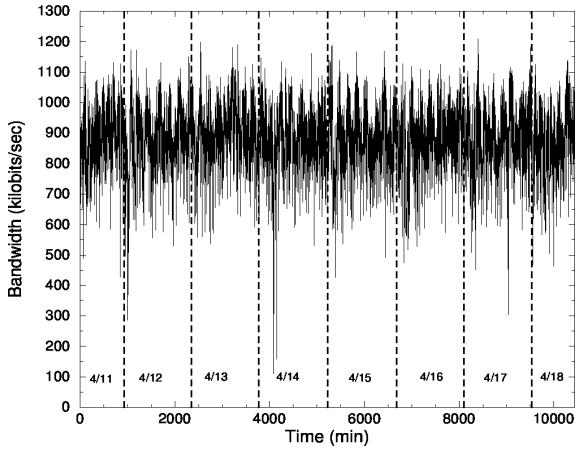
About the game...

- Centralized server implementation
  - Clients update server with actions from players
  - Server maintains global information and determines game state
  - Server broadcasts results to each client
- Sources of network traffic
  - Real-time action and coordinate information
  - Broadcast in-game text messaging
  - Broadcast in-game voice messaging
  - Customized spray images from players
  - Customized sounds and entire maps from server

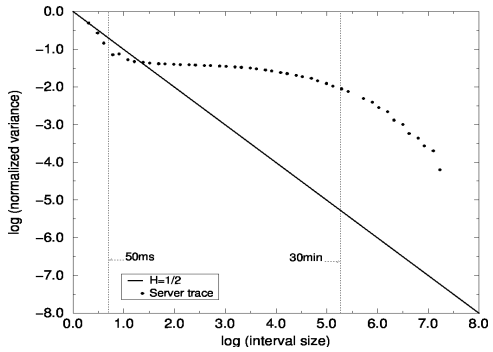
The trace

- cs.mshmro.com (129.95.50.147)
  - Dedicated 1.8GHz Pentium 4 Linux server
  - OC-3
  - 70,000+ unique players (WonIDs) over last 4 months
- One week in duration 4/11 – 4/18
- 500 million packets
- 16,000+ sessions from 5800+ different players

A week in the life...



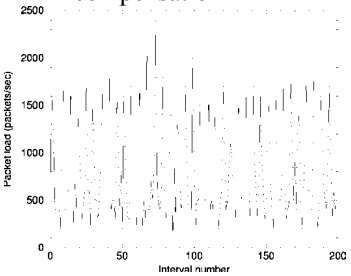
Variance time plot



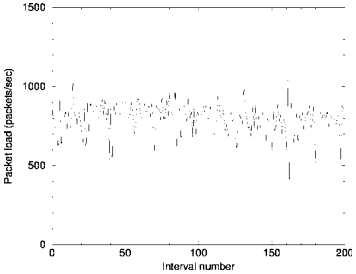
Normalized to base interval of 10ms

Digging deeper

- Periodic server bursts every 50ms
  - Game must support high interactivity
  - Game logic requires predictable updates to perform lag compensation



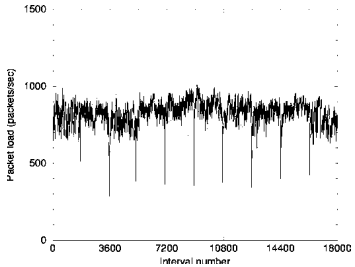
Interval size=10ms



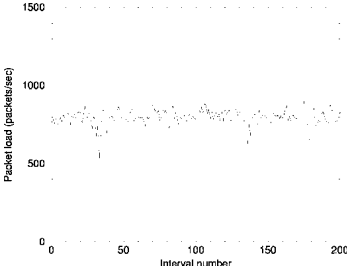
Interval size=50ms

Digging deeper

- Low utilization every 30 minutes
  - Server configured to change maps every 30 minutes
  - Traffic pegged otherwise....



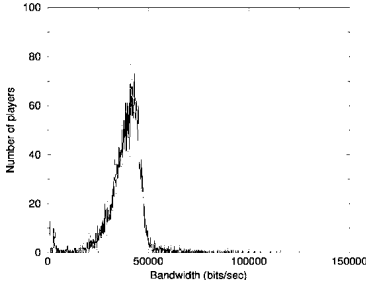
Interval size=1sec



Interval size=30min

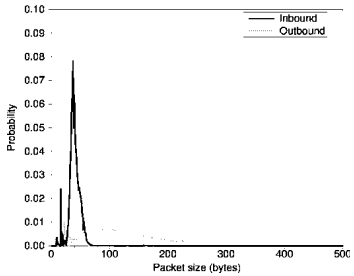
Finding the source of predictability

- Games must be fair across all mediums (i.e. 56kers)
  - Aggregate predictability due to “saturation of the narrowest last-mile link”
- Histogram of average per-session client bandwidth



Packet sizes

- Supporting narrow last-mile links with a high degree of interactivity *requires* small packets
  - Clients send small single updates
  - Servers aggregate and broadcast larger global updates



Implications

- Routers, firewalls, etc. must be designed to handle large bursts at millisecond levels
  - Game requirements do not allow for loss or delay (lag)
  - Should not be provisioned assuming a large average packet size [Partridge98]
  - If there are buffers anywhere, they must...
    - Use ECN
    - Be short (i.e. not have a bandwidth-delay product of buffering)
    - Employ an AQM that works with short queues

Implications

- ISPs, game services
  - Must examine “lookup” utilization in addition to link utilization
  - Concentrated deployments of game servers may be problematic
    - Large server farms in a single co-lo
    - America's Army, UT2K3, Xbox

On-going work

- Other pieces in the provisioning puzzle
  - Aggregate player populations
  - Geographic distributions of players over time (IP2Geo)
- Impact on route and packet classification caching
- Other FPS games
  - HL-based: Day of Defeat
  - UT-based: Unreal Tournament 2003, America's Army
  - Quake-based: Medal of Honor: Allied Assault
  - Results apply across other FPS games and corroborated by other studies

Future work

- Games as passive measurement infrastructure
  - Only widespread application with continuous in-band ping information being delivered (measurement for free)
  - "Ping times" of all clients broadcast to all other clients every 2-3 seconds
  - 20,000+ servers, millions of clients
- Games as active measurement infrastructure
  - Thriving FPS mod community and tools
  - Server modifications [Armitage01]

Questions?

Revisiting implications

- On-line games
  - Usage steadily increasing [McCreary00]
  - Fundamentally different than typical web traffic
  - Large, periodic flows
  - Has a detrimental affect on routers
- Question: How do emerging traffic characteristics affect packet classification caching mechanisms?

Route caching

- Used currently in IP destination-based routing
  - One-dimensional classifier
  - Avoid route lookups by caching previous decisions
  - Instrumental in building gigabit IP routers

Previous caching work

- Cache of 12,000 entries gives 95% hit rate [Jain86, Feldmeier88, Heimlich90, Jain90, Newman97, Partridge98]
- "A 50 Gb/s IP Router" [Partridge98]
  - Alpha 21164-based forwarding cards (separate from line cards)
    - First level on-chip cache stores instructions
      - Icache=8KB (2048 instructions), Dcache=8KB
    - Secondary on-chip cahe=96KB
      - Fits 12000 entry route cache in memory
      - 64 bytes per entry due to cache line size
    - Tertiary cache=16MB
      - Full double-buffered route table

Packet classification caching

- Multi-field identification of network traffic
  - Typically done on the 5-tuple
  - `<SourceIP, DestinationIP, SourcePort, DestinationPort, Protocol>`
  - Inherently harder than Destination IP route lookup
  - Extremely resource intensive
- Many network services require packet classification
  - Differentiated services (QoS), VPNs, MPLS, NATs, firewalls

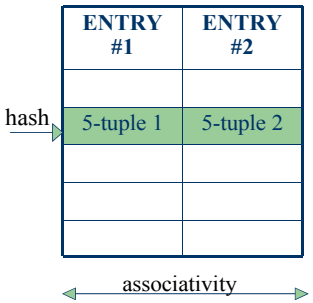
Packet classification caching

- Overhead of full, multi-dimensional packet classification makes caching *even more important*
  - Full classification algorithms much harder to do versus route lookups
  - Per-flow versus per-destination caching results in much lower hit rates
  - Rule and traffic dependent

Goal of study

- Attack the packet classification caching problem in the context of emerging traffic patterns
- Resource requirements and data structures for high performance packet classification caches
  - What cache size should be used?
  - How much associativity should the cache have?
  - What replacement policy should the cache employ?
  - What hash function should the cache use

General cache architecture



Current approaches

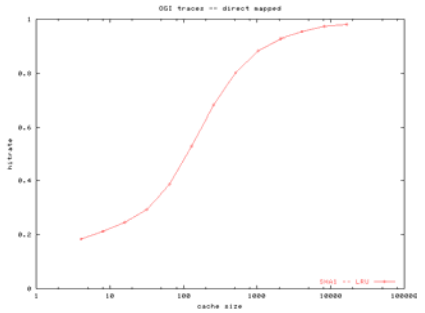
- Direct-mapped hashing with LRU replacement
  - Typical for IP route caches [Partridge98]
- Parallel hashing and searching with set-associative hardware [Xu00]
  - ASIC solution with parallel processing and a fixed, LRU replacement scheme

Approach

- Collect real traces
  - http://pma.nlanr.net
  - OGI/OHSU OC-3 trace
- Simulation
  - PCCS
- Real Hardware tests
  - IXP1200

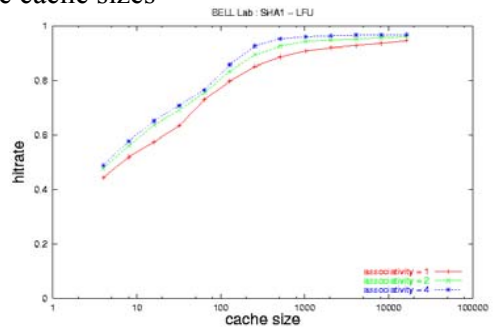
How large should the cache be?

- Depends on number of simultaneously active flows present (assuming each new flow has a new 5-tuple)



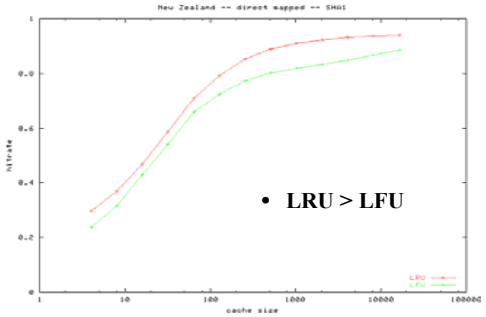
What degree of associativity is needed?

- Associativity increases hit rates
- Benefits diminish with increasing associativity and large cache sizes



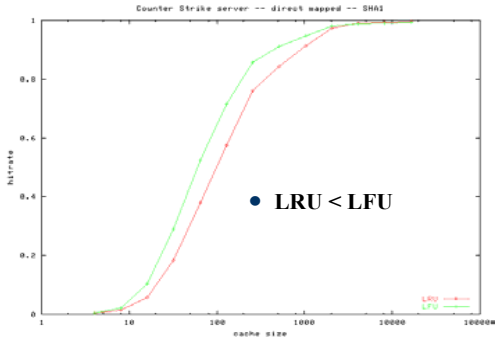
What replacement policy is needed?

- LRU: Least-recently used
- LFU: Least-frequently used



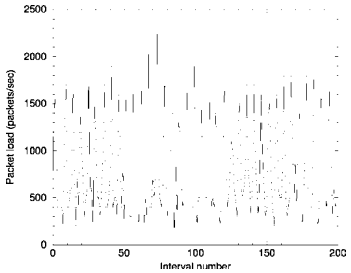
- **LRU > LFU**

What replacement policy is needed?



- **LRU < LFU**

Recall game server traffic

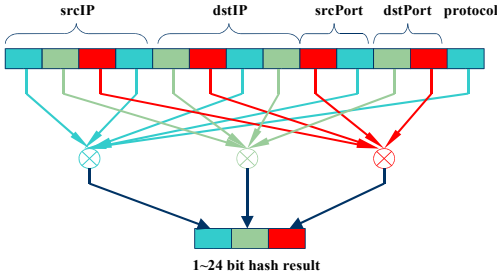


Observations

- Game traffic
  - Large number of periodic packets
  - Extremely small packet sizes
  - Persistent flows
  - Without caching, a packet classification disaster
- Web traffic
  - Bursty, heavy-tailed packet arrival
  - Transient flows
- Consider a mixture of game and web traffic
  - LFU prevents pathologic thrashing

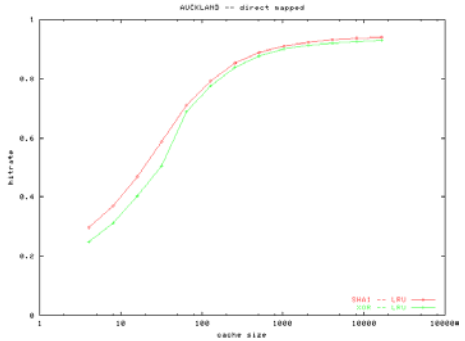
What hash function is needed?

- IP address and address mixes highly structured
  - Strong hash functions prevent collisions
  - Weak hashing leads to increased thrashing and misses
- Compare SHA-1 with "dummy" hash function



What hash function is needed?

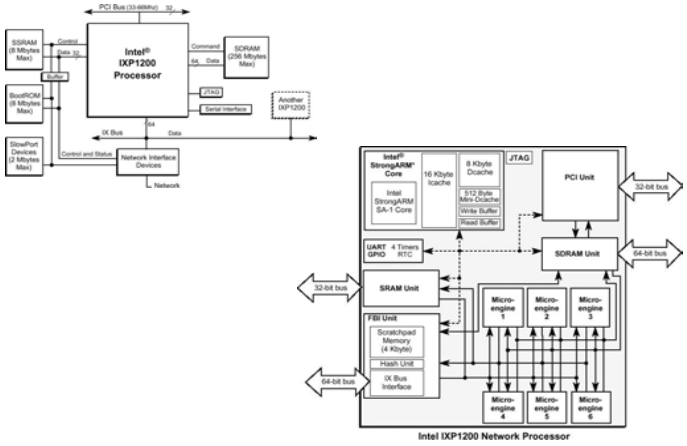
- Weak hash function performs reasonably well
- Hardware hash units not needed for caching



Experimental validation

- Intel IXP1200
  - Programmable network processor platform
  - Can be used to explore sizing, associativity, and hashing issues
  - Provides a single 64-bit hardware hash unit
    - Fixed multiplicand polynomial
    - Programmable divisor polynomial
- Question: Should the IXP's hash unit be used to implement a packet classification cache?

IXP1200

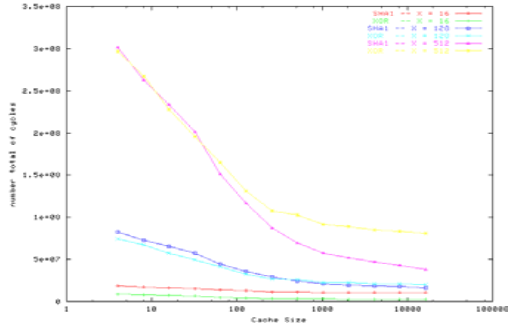


IXP performance tests

- Hash unit performance test implemented in microC
  - Latency ~ 25-30 cycles
  - Throughput ~ 1 result every 9 cycles
- Dummy hash function
  - Latency ~ 5 cycles
  - Throughput ~ 1 result every 5 cycles per micro-engine
- Compare total number of cycles taken assuming a cache miss incurs a full packet classification taking t_X cycles

Results

- h=hit rate t_h=hash latency t_X=classification latency
- Total cycles = $h * t_h + (1-h)*t_X$



Conclusion and future work

- Emerging on-line games create a workload that is hard for current infrastructure to handle
- Network hardware must be provisioned accordingly
- Network hardware designs such as caches must adapt to changing traffic structure
  - Cache sizes, associativity, replacement policies, hash functions
- Future work
  - Understanding the evolution of on-line games
  - Examining additional architectural features for improving performance