Origins of NVE

The author of these slides is Dr. Mark Pullen. Students registered Dr. Pullen's course may make a single machine-readable copy and print a single copy of each slide for their own reference, so long as each slide contains the copyright statement. Permission for any other use, either in machine-readable or printed form, must be obtained from the author in writing.

Lecture Overview

- DoD NVE
  - SIMNET and its architecture
  - Distributed Interactive Simulation (DIS)
  - distribution, heterogeniety, scalability of DIS
- Networked games/demos
  - SGI Flight and DogFight
  - Doom
- Academic NVEs
  - NPSNET
  - PARADISE
  - DIVE
  - Bricknet
  - MR Toolkit

Department of Defense NVE

SIMNET

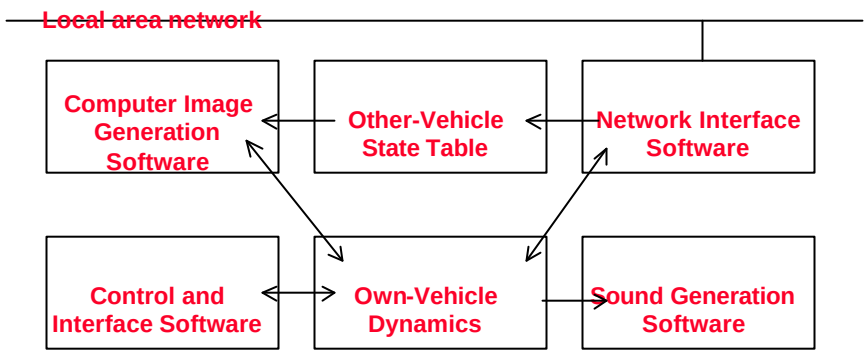
- a DARPA research project that went into production
- research questions:
  - how to fabricate high-quality low-cost simulators
  - how to network them to create a consistent, virtual battlefield

SIMNET

SIMNET Architecture

- object-event
  - 1 object per simulator, one event per message
- autonomous simulation nodes with embedded dead reckoning
  - (send message only when the predictor fails - small jump to correct)
- terrain separate from objects- cannot be changed
- heartbeat updates state 1 pdu/s
  - stationary bit turns off dead reckoning
  - Ethernet multicasting (no WAN protocol)

SIMNET Architecture



SIMNET Appearance Packet

Field Name	*Contents*	*Bytes*
Vehicle ID	Site-Host-Vehicle	6
Vehicle class	Tank/Simple/Static/Irrelevant	1
Force ID		1
Guises	Object description	8
World coordinates	x,y,z	24
Rotation matrix		36
Appearance		4
Markings	text field	12
Timestamp		4
Capabilities		32
Engine speed		2
Stationary bit/padding		2
Vehicle appearance variant	velocity,turret azimuth, gun elevation	24

SIMNET Dead Reckoning

- To reduce packet traffic in SIMNET, the objects and ghosts paradigm was created. The idea behind this paradigm is that objects only place packets onto the network when their home node determines that the other nodes on the network are no longer able to predict their state within a certain threshold amount.
- This paradigm assumes that the other nodes in the system are maintaining “ghost” copies of the object in their memories and that the last reported direction, velocity, and position are sufficient to predict, within the threshold amount, where that entity is now.

SIMNET Results

- scaled to 850 objects in WARSIM exercise
- eventually 11 sites (4 in Europe)
  - not interconnected
- Grafenwoehr use for Canadian gunnery trophy preparation credited with turnaround in skills

Department of Defense NVE

Distributed Interactive Simulation (DIS)

IEEE standard intended to make SIMNET results replicable

same basic object-event/autonomous simulator/DR architecture as SIMNET

- 4 basic PDUs: entity state, fire, detonation, collision
- 23 more for simulation control, EM emanations, support
- 5 second heartbeat threshold
- requires a fair simulator (physics modeled)

DIS Protocol Data Units (PDUs)

- Entity-state PDU (ESPDU)
  - all data needed to represent an object
- collision: supposed to use physics of impact
  - but more often models destruction
- fire & detonation: receiver changes visualized object
  - because of UDP network, packets may be lost
  - different state may be seen at various nodes
  - argument: this chaotic behavior models "fog of battle"
- observed statistics:
  - ESPDU 96% fire .16%
  - detonation .16% collision .04%

DIS Entity-State PDU

Field Name	*Contents*	*Bytes*
Header	version, exercise ID, PDU type	12
	protocol family, time stamp, length	
Entity ID	site, application, entity	6
Force ID		1
Number of articulations		1
Entity type	kind, domain, country, category,	8
	subcategory, specific, extra	8
Linear velocity	x,y,z	12
Location	x,y,z	24
Orientation	ψ,θ,ϕ	12
Appearance		4
Dead reckoning	algorithm, other parameters,	40
parameters	linear acceleration, angular velocity	
Marking	character set, text	12
Capabilities		4
Articulation parameters	type, change, ID, attached to. etc.	n*16

Distribution, Heterogeniety, Scalability of DIS

- any computer on the net that reads/writes DIS can participate
- dynamic terrain and environmental effects are impractical computationally
- defines 9 DR algorithms protocol-selectable
- much constant data in the PDUs - Cohen suggested send only what changes (see SRMP later)
- designed for small engagements (300 objects) whereas DoD would like 100K
- RITN made DIS work at 5000 objects (for 3 to 5 minutes) - hacked up architecture
- not extensible except through "experimental PDU" (which has been used a lot)

Networked Games/Demonstrations

SGI Flight and DogFight

- Flight: Blue Angels to show off SGI graphics (great sales tool 1984-1992)
- beginning 1984, networking was added to Flight in stages
- 1985, multi-Flight became Dogfight
  - multicast at frame rate: ate people time and net capacity
- may have inspired more NVEs than SIMNET & DIS

Doom

- PC game (1993), shareware giveaway:
  - 15 million downloads; 250,000 registered users
- congested LANs by sending at frame rate
- believable 3D sparked interest resulting in many other games
- Doom II (1994)1.6 m copies sold
- many others followed: Marathon & Bolo (for Macintosh)
  - Bolo limited network impact by using a virtual ring of hosts
  - latency increases with number of participants

Academic NVEs

Goal:
get the technology into the academic community

NPSNET

longest continuing academic NVE effort

- early graphics work on SGI 1986
- FOG-M missile simulator ran over Fort Hunter Liggett
  - SGI .5 MIPS Unix; not networked
- VEH developed from videotape of actual missile
  - limited networking (blocking read/write)
- 3 generations of MovingPlatform Simulator (MPS)
  - 5 workstations on LAN with NPS protocols
- NPSNET 1 & 2 experimented with SIMNET ideas
- NPS Stealth ran SIMNET protocols on workstation
  - interoperation with SIMNET $350K simulator
- NPSNET-IV converted to DIS 1993 - ran over MBone
- NPSNET-V overhaul in 2002

NPSNET Evolution

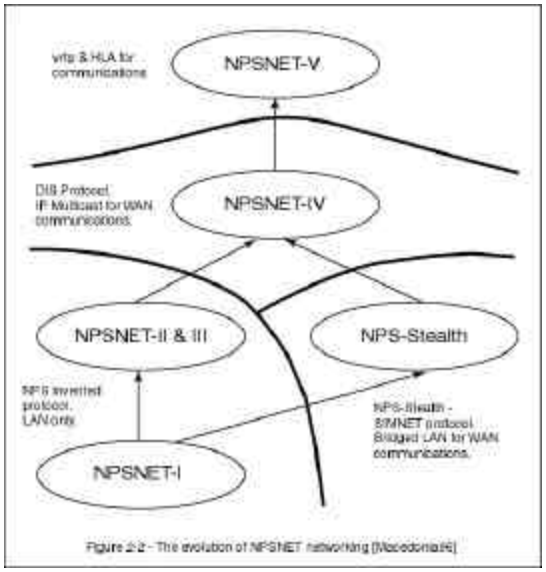


Figure 2-2 - The evolution of NPSNET networking [Macedonia96]

NPSNET WAN

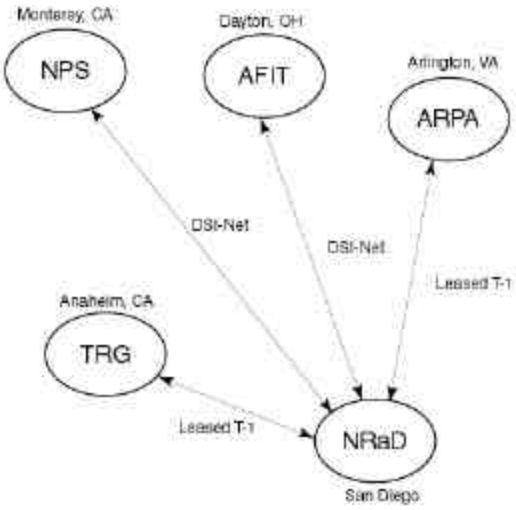


Figure 2-3 - NPSNET connectivity at SIGGRAPH 93 [Macedonia dissertation]

Performance Architecture for Advanced Distributed Simulation Environments

PARADISE

Singhal and Cheriton at Stanford

- started as dynamic terrain for SGI Dogfight
- addressed network issues; used multicast emulator
- everything is an object, with area of interest (AOI)
- improved DR: position History-Based Dead Reckoning (PHBDR) - smaller update packets
- multiple independent communication flows per object
  - allows DR at different levels of accuracy
- reliable multicast protocol
  - Log-Based Receiver-Reliable Multicast
- terrain server for dunamically paging terrain over the network
- C++ modules plugged together to form an entity; Tcl interface
- 50 to 70 entities on four RS/6000 WS

Distributed Interactive Virtual Environment (DIVE)

Swedish Institute of Computer Science

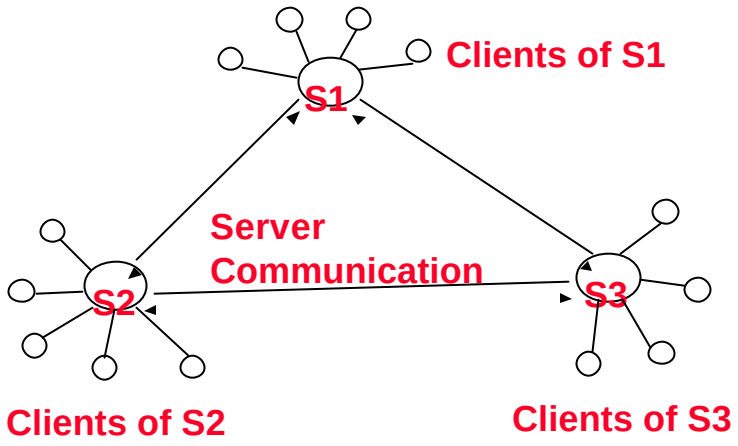
- ISIS toolkit process groups
  - simulate large shared memory over network
  - set of processes addressed as single entity via MC
  - difficult to scale beyond 16 to 32 participants
- entire database is dynamic
  - add new objects in reliable/consistent way
- application: collaboration and interaction
  - simulated conference room with human avatars
- started as telepresence project
  - shared object via networked workstations (e.g. clock)
- this was DIVE 2; next DIVE 3 used Floyd's SRM and Tcl

BrickNet

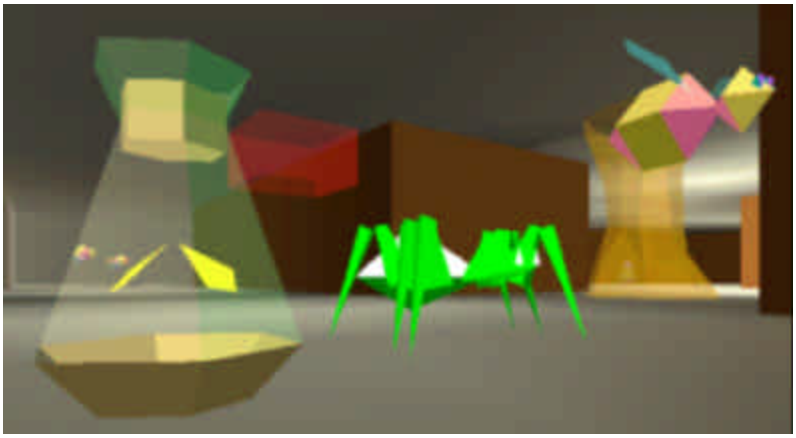
Singh at National University of Singapore

- partitions virtual world among VE clients
- communication mediated by servers (object request broker)
- intended for collaborative design (up to 32 participants)
  - each server could be a different virtual world
- has been used for games, groupware systems, concurrent engineering
- servers are bottlenecks
- not replicating the database is unusual
- extended to behavior sharing
  - this made it better for games (CyberBug)
- spun off to PC at NetEffect (large low-end VEs)
  - used for HistoryCity app (based on history of Singapore)

BrickNet Client-Server Communication



BrickNet CyberBugs



MR Toolkit

Minimal Reality Toolkit Peer Package

- UDP-based application-layer multicast
  - frequent refresh
  - no DR
  - lower delay
- local caching
- fully connected mesh
  - N^2 messages for N participants
  - does not scale - limit 4 participants