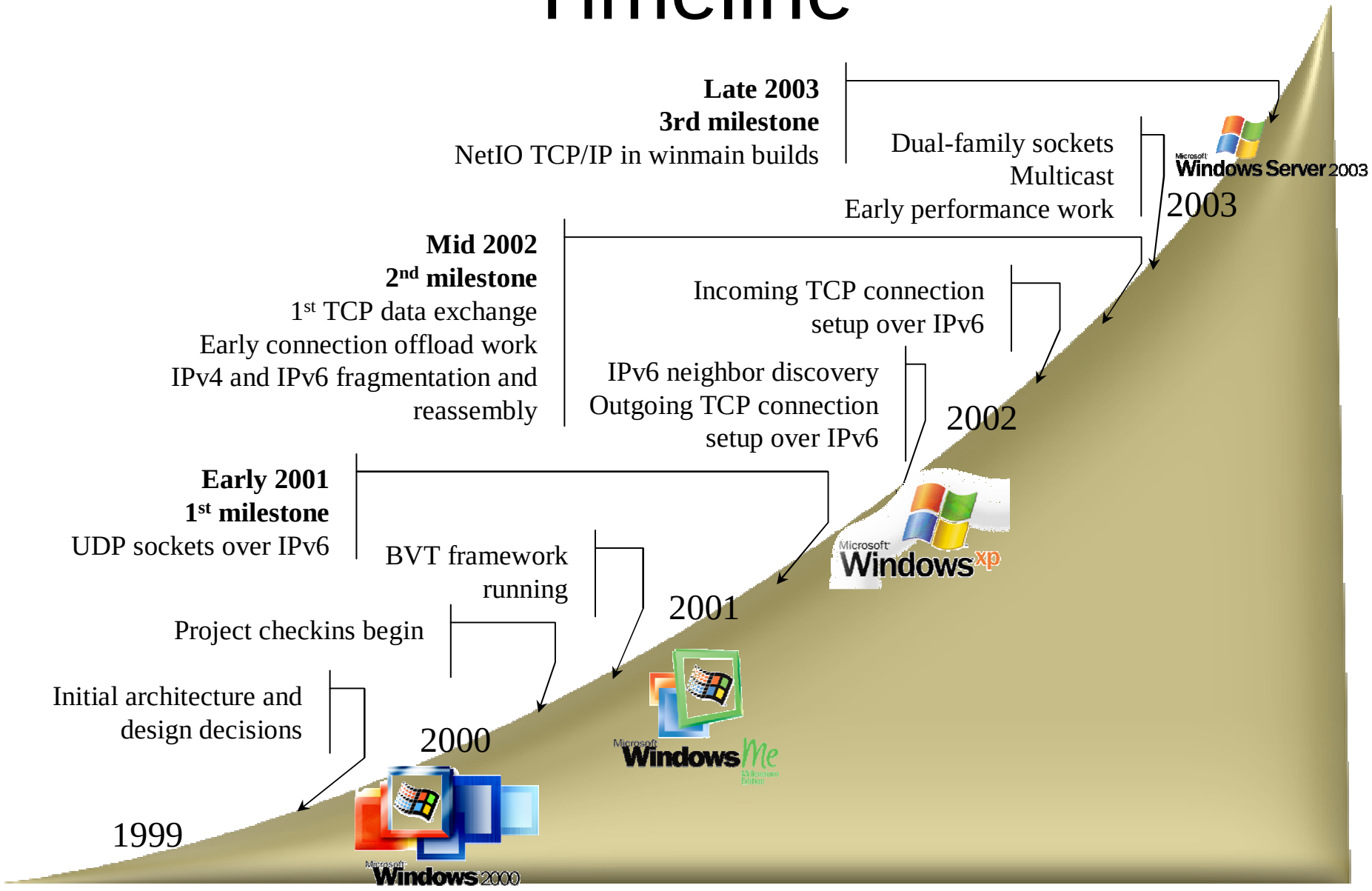
The NetIO stack in Windows Vista: functionality & deployment

Abolade Gbadegesin
Architect
Microsoft Corporation

Getting Started

- About me
  - Responsible for architecture of networking transports in Windows
  - 11 years working on Windows networking
  - 6 years redesigning and rewriting the Windows networking stack
- About this talk
  - New and significantly updated out-of-box behavior
  - Additional latent capabilities
  - What we've learned, and what we're figuring out how to learn
  - Where we go from here

Timeline


Late 2003
3rd milestone
NetIO TCP/IP in winmain builds
Dual-family sockets
Multicast
Early performance work
Windows Server 2003
2003
Mid 2002
2nd milestone
1st TCP data exchange
Early connection offload work
IPv4 and IPv6 fragmentation and reassembly
Incoming TCP connection setup over IPv6
IPv6 neighbor discovery
Outgoing TCP connection setup over IPv6
2002
Windows xp
Early 2001
1st milestone
UDP sockets over IPv6
BVT framework running
2001
Windows Me
Project checkins begin
Initial architecture and design decisions
2000
Windows 2000
1999


Meet the Internet Protocols Team

New and Updated Behavior

- IPv6 on by default [6to4, ISATAP, Teredo, DHCPv6, DNS]
- Equal-cost multipath routing support
- Dead-gateway & black-hole router detection on by default
- TCP negotiated options: window scaling, SACK
- TCP receive window auto-tuning
- Compound TCP
- NewReno, F-RTO, D-SACK support
- IANA compliance for ephemeral ports
- Numerous configuration settings replaced with self-tuning behavior

Latent Capabilities

- ECN configurable
- Strong host model by default (weak host configurable)
- Link-Local Multicast Name Resolution (LLMNR)
- Per-socket TCP extended statistics
- TCP connection offload
- Support for attempting IPsec on all traffic with seamless fallback to clear
- Policy-based QoS for Active Directory domains
- Integrated network diagnostics framework

Case Study #1: Enabling IPv6 By Default

- Phase 1: hosts
  - Experimental bootstrapping (Windows 2000 add-on)
  - Early self-hosting (Windows XPSP2 & Windows Server 2003
  - Public 6to4 and Teredo services
- Phase 2: infrastructure and services
  - Native IPv6 and ISATAP routers
  - IT-supported dual-stack deployments
- Phase 3: widespread deployment
  - On-by-default for all nodes (Windows Vista Beta 2 and Windows Server Code-Name Longhorn Beta 2)
  - IPv6-only mode for selected corporate desktops

Case Study #1: Lessons

- Root out the hard problems early
- Pick the right questions and build mechanisms to help answer them as part of deployment
- Questions we've had to answer:
  - What are the minimum host and infrastructure mechanisms needed to enable IPv6 in a broadly-available consumer OS? In managed networks? In public networks?

Case Study #2: DNS AAAA Queries

- Summary: IPv6-capable nodes ask for both A and AAAA records
  - DNS devolution compounds number of queries
- Impact on MSFT self-host deployments
  - Minimal; most clients query AD DNS servers
  - Negligible increase in load
- Impact on public deployments
  - MSCOM: No measurable impact (yet)
  - NTT Labs: NANOG report with preliminary analysis based on Windows Vista Pre-Beta2 CTP (Build 5270)
  - Dan Kaminsky: http://www.doxpara.com/?q=node/1136

Case Study #2: Status & Lessons

- Status in Vista RC1 (when global IPv6 address is present):
  - DNS sends A query first, follows up with AAAA only to servers that have some info, then stops
    - Much less than 2x increase in DNS traffic from nodes with a global IPv6 address
  - Most nodes will have a global IPv6 address only when an application is using IPv6
    - Vista doesn't send AAAA queries if the only global IPv6 addresses it has are Teredo addresses
    - Teredo configures global IPv6 address 'just-in-time'
  - http://www.microsoft.com/technet/itsolutions/network/ipv6/vista_dns.mspx
- Lessons:
  - ❖ Deployments of new behavior are best undertaken as joint efforts between host software vendors and public network operators

Case Study #3: TCP Receive Window Auto-tuning

- The Challenge:
  - ❖Can end-to-end TCP performance for end-users be significantly and broadly improved, without impairing the capacity planning and bandwidth management mechanisms that network operators rely upon?

- Design:
  - – Tune window advertisements based on receive-side bandwidth estimates

Case Study #3: Status

- Results
  - MSCOM: 40x out-of-box improvement in raw TCP throughput
- Feedback
  - Microsoft IT: need to prioritize certain traffic on corporate network
- Responses
  - Added Group Policy controls over TCP auto-tuning
  - Evangelize Policy-based QoS for managed networks

Case Study #4: Compound TCP

- The Challenge:
  - ❖ Can a hybrid TCP be broadly deployed alongside classic TCP flows, preserving fairness while ramping up faster from losses?

- Design
  - Combine delay-based congestion window with NewReno loss-based congestion window
  - Enable on paths with sufficient buffering to facilitate delay sensing
  - Configurable in Windows Vista, currently on by default in Windows Server Code-Name Longhorn

Case Study #4: Results

- MSCOM
  - 50% out-of-box reduction in transfer time
- UK Ministry of Defense
  - IET conference report finding Compound TCP achieved 2x faster recovery from errors over satellite links
  - http://www.iee.org/oncomms/pn/satellite/10%20-%20Race%20&%20Thomas.pdf
- Stanford Linear Accelerator Center
  - Technical report finding Compound TCP more fair to classic TCP than HighSpeed-TCP
  - http://www.slac.stanford.edu/pubs/slactns/slac-tn-06-005.html

Next Steps and Proposals

- Other challenges and opportunities could benefit from collaboration between software vendors & operators
  - IPsec-everywhere
  - Peer-to-peer traffic patterns
  - Botnets and distributed denial of service
- Proposal:
  - Generate list of ‘open questions’, starting with those related to Windows Vista
  - Determine necessary metrics across hosts and infrastructure
  - Assess progress and recalibrate on an ongoing basis

discussion

feedback: tcpipfb@microsoft.com