NadaNet Networking for Apple II Programmers

Michael Mahon

kansasfest 07

apple II

Why NadaNet?

- In the early 1990s, I became interested in “clustered” machines: parallel computers connected by a LAN.
- This interest naturally turned to Apple II computers, and the possibility of an all-software Apple II network.
- Preliminary work indicated that data rates of several kilobytes/second were possible using a digital phase-locked loop, adequate for an Apple II cluster.
- It seemed like a great opportunity to “reinvent” Ethernet and learn from the bottom up!

What is NadaNet?

- A peer-to-peer baseband serial bus network
- Implemented entirely in Apple II software
- Both Applesoft and M/L APIs
- Contention-based, with collision avoidance (CSMA/CA)
- Connecting dozens of Apple II's
- >10K bytes/second bandwidth
- 1 to 256-byte packets
- 1 to 64K-byte atomic data transactions

A network to support parallel programming *on Apple II's*.

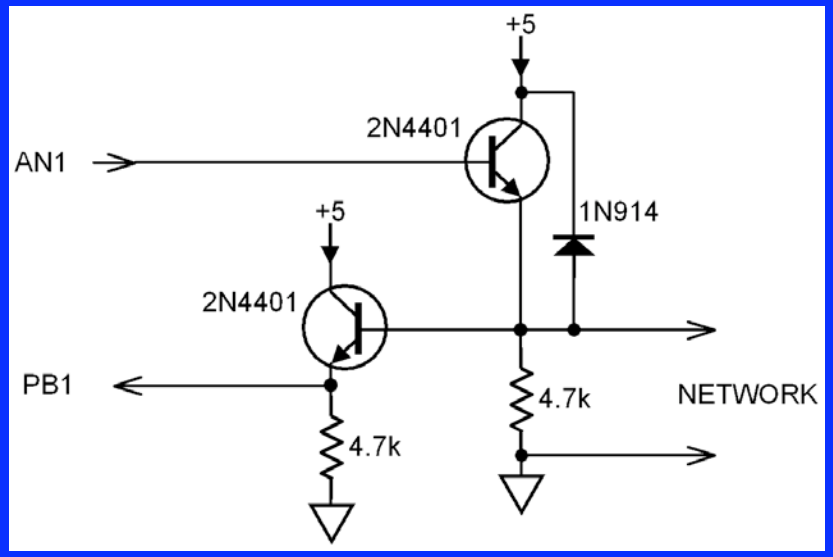
What NadaNet is Not

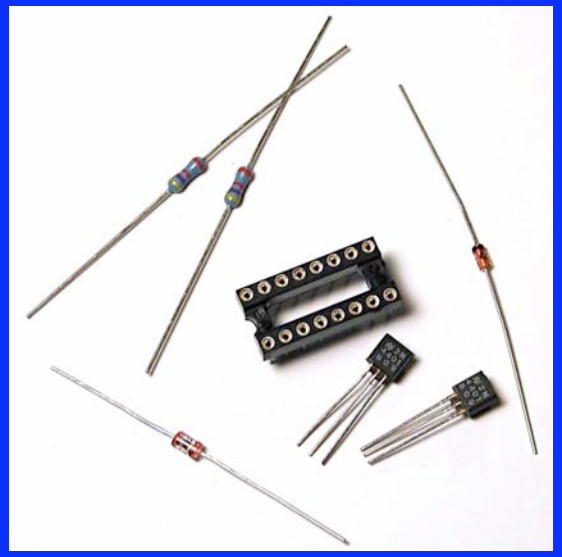
- A way of connecting to the Internet
- A way of connecting to non-Apple II's
- Compatible with TCP/IP networks

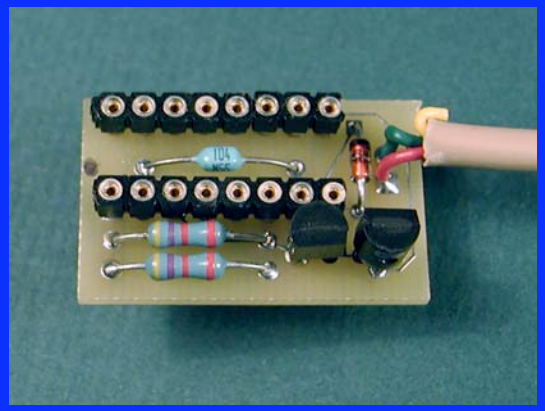
NadaNet Requirements

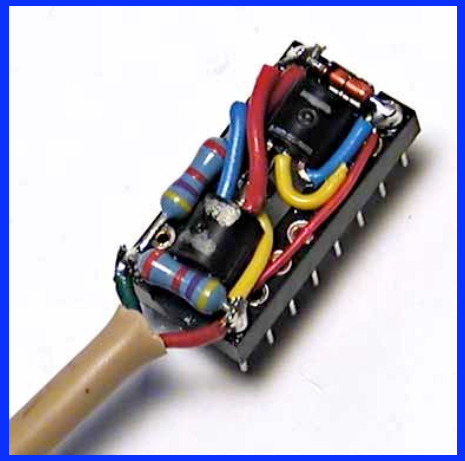
- Two or more Apple II's
- Each machine requires a game port adapter
  - 16-pin internal port (**not** 9-pin joystick port)
  - The Apple II and II+ allowed a short network without an adapter, but later machines present heavier loads.
- Adapters are simple and can be built easily
- Adapters extend the physical length of the network
  - 250 feet if shielded cable used
  - 500 plus feet if Cat 5 UTP used

NadaNet Adapter

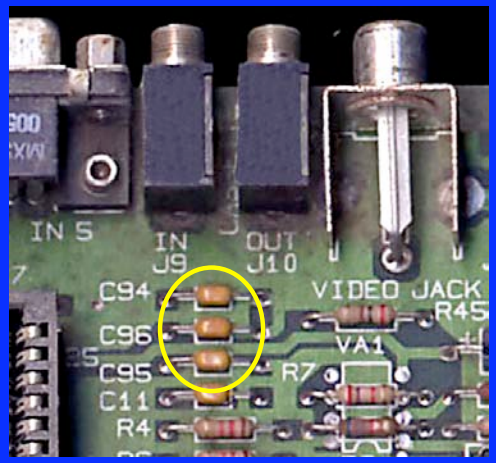








Fixing late-model //e's



- Just snip one (or both) leads on each capacitor to restore full bandwidth capability to pushbutton inputs.
- Can snip one lead and lift cap slightly for minimal mod.
  - Can always be re-soldered to handicap your Apple!
- 0.1 uF capacitors to ground were added in "Platinum" board revision as a brute-force EMI fix.

How is it used?

- NadaNet is loaded and initialized
  - BLOAD from disk and initialize, usually by HELLO or STARTUP
  - (or boot from network)
- Machines are only “on” the net when they are “serving”--listening for packets and acting on them.
- Any machine can process requests while it is serving.
- Any machine can make requests at any time, but they will only be acted upon if the target machine is serving.
  - Default request timeout is ~3 seconds.
- If a Message Server is serving, messages can be sent and received at any time.

NadaNet Request APIs

- &PEEK (dest,addr,len16,locaddr)
- &POKE (dest,addr,len16,locaddr)
- &CALL (dest,addr [,ax])
- &PUTMSG (msid,mclass,mlen8,maddr)
- &GETMSG# (msid,mclass,mlen8,maddr)
- &PEEKINC (dest,addr,incval16,oldval16)
- &BPOKE (addr,val16)
- &SERVE# (iter)
  - Until 1 request, iter*20ms timeout, or key is pressed
- CALL 973
  - Serves indefinitely

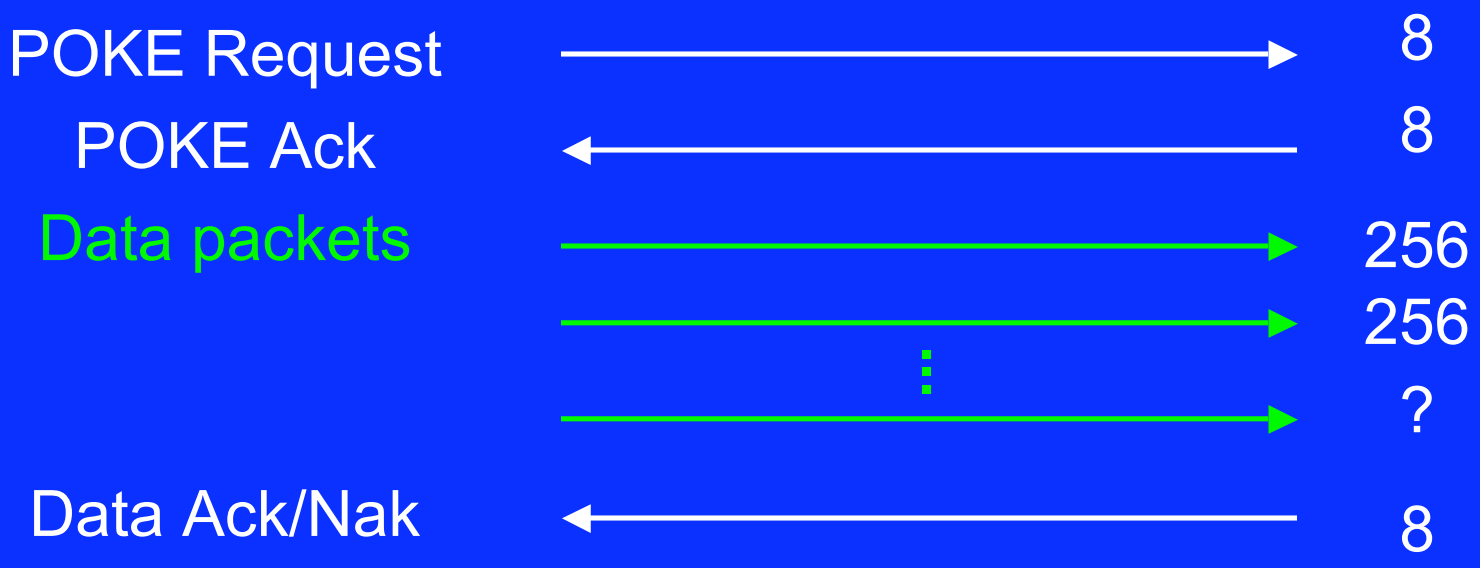
Machine 10 Demo Program

```
100 REM NadaNet Hi-res Demo
110 REM MJM - June 23, 2007
120 :
130 D$ = CHR$(4)
140 PRINT CHR$(21) : REM 40-Col mode
150 HGR2
160 PRINT D$"BLOAD NADAHGR"
170 CALL 973 : REM Serve "forever"
```

Examples

- &PEEK (10,2*4096,2*4096,2*4096)
  - Moves $2000 bytes from machine 10's HGR1 buffer to our HGR1 buffer, where we can see it.
- &POKE (10,4*4096,2*4096,2*4096)
  - Moves $2000 bytes from our HGR1 buffer to machine 10's HGR2 buffer, replacing its contents.

POKE: A Typical Protocol



Optimization:

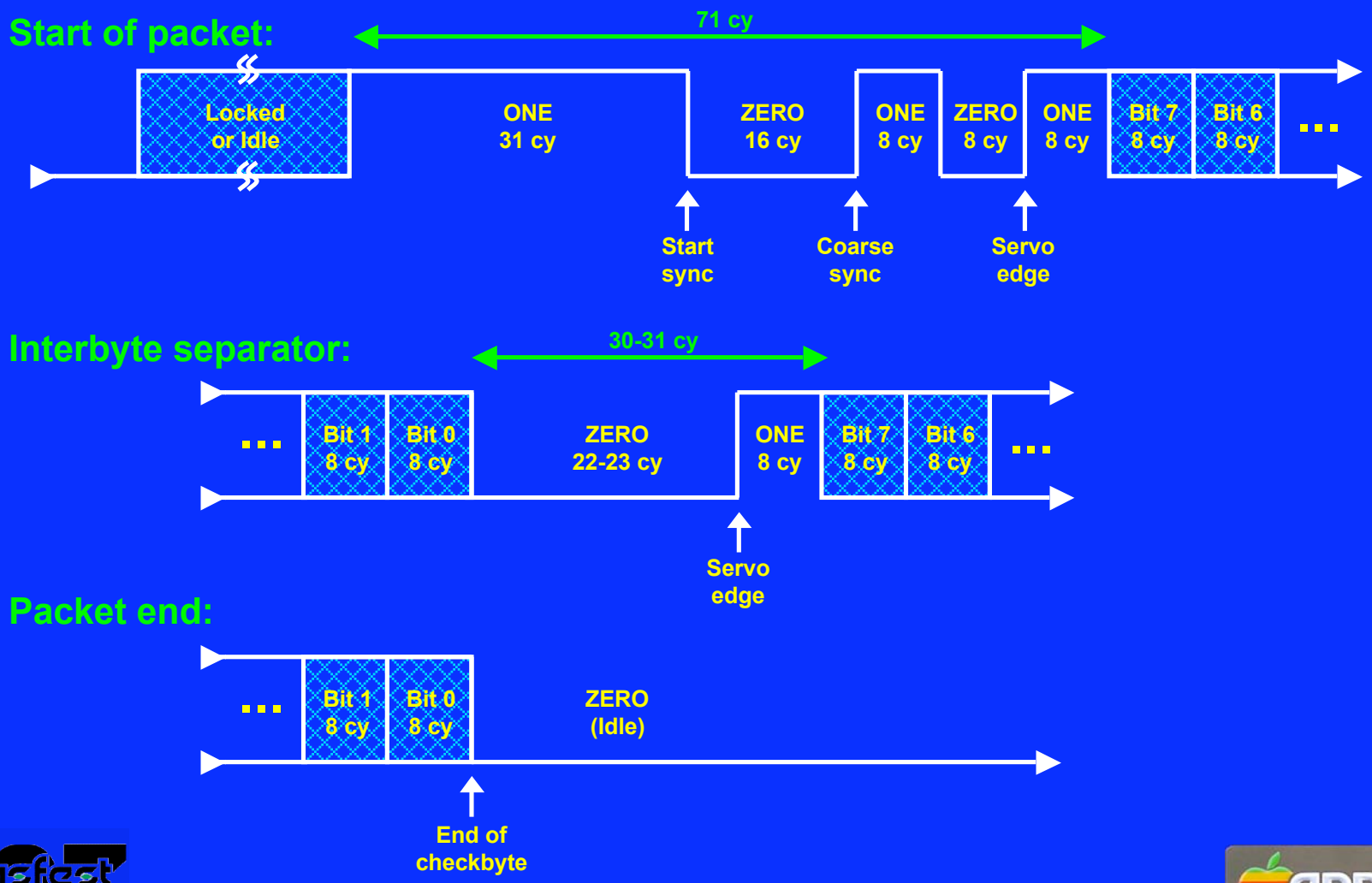
Data packets and Data Ack are omitted if length is 1 or 2 bytes.

Control Packet Format

- Request identifies all control packets of a given request type
  - PEEK, POKE, CALL, etc.
- Modifier specifies the role of the packet in the protocol
  - Request, Request Ack, Data Ack, Nak
- Dest identifies the target machine
- From identifies the sending machine
- Address (generally) specifies a target machine address
- Length (generally) specifies a data length
- Cksum is an EOR checksum applied to all packets

Control packets are ~1ms long.

Nadanet Data Format



NadaNet Arbitration

- Always listen before sending
- Wait for net to be idle for 1 millisecond + id * 22cy
  - Lower id machines have higher arbitration priority
- Seize net by forcing ONE state

- Consequences:
  - Network is “locked” until it is idle for longer than 1ms.
  - All requests meet this requirement and so are atomic.

AppleCrate

- An 8-machine Apple //e cluster
- ROMs modified for NadaNet boot (from server)
- Powered by PC power supply

Questions and discussion...