# Router Tests V.4

Scott Bradner

Harvard University

## Testing the Devices

Just how does one find out
what is real?

Scott Bradner
Harvard University
sob@harvard.edu



## BMWG

Benchmarking Methodology Working Group

- part of IETF
  Internet Engineering Task Force
- sanctioned by IAB
  Internet Activities Board

- WG is addressing:
  1 - define terms (RFC 1242)
    throughput, latency, etc
  2 - define device classes
    stand alone
      bridges, routers, gateways
    host dependant
      network interfaces, software
  3 - define specific tests
  4 - define reporting terminology

- Mailing list
  bmwg-request@harvard.edu



## Life on a real world network:

pathological conditions:

peak load
- arp storms
- broadcast storms
- rwho on diskless nodes
- bootp
- tftp booting
- graphics
- backbone
- star network design

bursts of packets
- NFS traffic
- routing updates



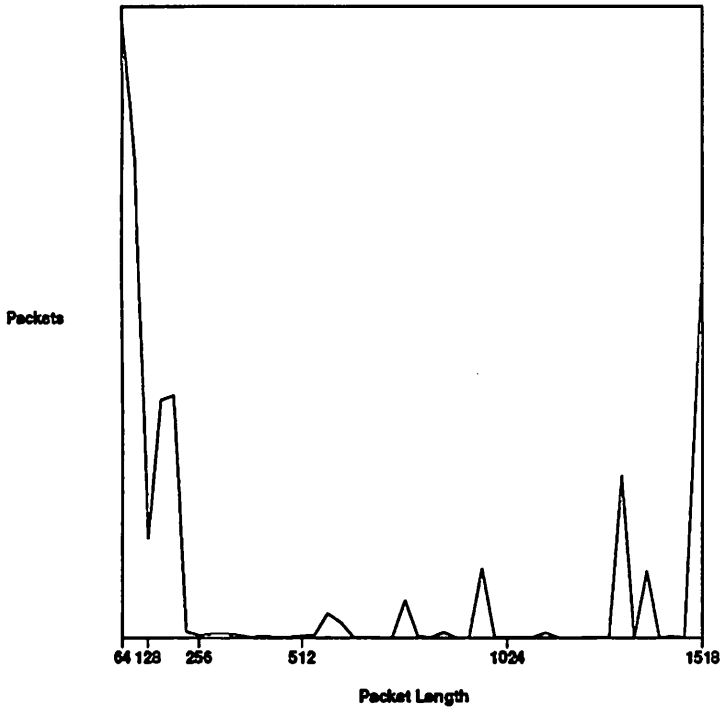
## Life on a real world network:

"normal" conditions

- NFS servers
- named
- NNTP
- SMTP
- PC clusters
- FTP
- terminal servers

## Life on a real world network:

packet size distribution at Harvard






## Other things to consider:

network management
- standards
  SNMP, CMOT
- proprietary

documentation
- fit for human consumption?

user interface
- how expensive a guru is needed?

reachability
- can it be managed over the network?
- how easy is it to crash the router so that it requires manual intervention?
- how easy is it to overload the router so that the processor does not respond to commands on the serial line?



## More things to consider:

rebooting requirments
- when do you have to reboot the box?
- how long does it take?

security
- what access controls on router?
- what sorts of filtering can be done on traffic?
  - on source of traffic
  - on destination of traffic
  - on protocol type?

network design
- pools linked with WAN lines
- 236 Ethernets into a box



## Testing, how to simulate real world

- can't do a very good job of simulating the "real world"

- easy to check simple things
  idle state
  delay through router
  effects of various filtering options
  accuracy of counters
  reaction to error packets
  effect of different protocols

- not too hard to simulate the pathological conditions
  high offered load
  back to back packets
  bidirectional traffic
  mixed protocols

- much harder to test for table space related limits
  routing table size
  arp cache size
  filtering list space

## Tests - background

- packet size
  from RFC 1242:

  > The number of octets in the frame from the first octet following the preamble to the end of the FCS, if present, or to the last octet of the data if there is no FCS.

- sizes used
  64, 128, 256, 512, 768, 1024, 1280, 1518
  (AppleTalk - 64, 128, 256, 512)

- protocols used:
  bridge test packet - protocol 9000x
  IP - UDP Echo Request
  AppleTalk II - AppleTalk Echo Request
  IPX - Echo Request
  DECnet Phase IV - short data packet

- WAN info
  56KB & T1 (1.536MB) used
  max rate set to 110% of calculated max



## Tests - throughput

- definition from RFC 1242:

  > The maximum rate at which none of the offered frames are dropped by the device.

- performed on Ethernet

- procedure:
  send "learn" packets
  for each packet size
    set *rate* = 1/2 maximum
    loop:
      send 30 sec burst at *rate*
      if all packets received
        increase *rate*
      if not all packets received
        decrease *rate*
      detect if thrashing



## Tests - packet loss rate

- definition from RFC 1242

  > Percentage of frames that should have been forwarded by a network device under steady state (constant) load that were not forwarded due to lack of resources.

- performed on Ethernet, FDDI and token ring

- procedure - Ethernet
  send "learn" packets
  for each packet size
    set *rate* to 100% of max rate
    loop:
      send 30 sec burst
      record passed packets
      reduce *rate* to next "10%"
        100% to 90% to 80% ...
      if 2 successive trials with all passed
        goto next packet size

- procedure - FDDI and token ring
  for each packet size
    send specific number of packets at test rate
      record number of packets passed
    reduce rate to next test rate
      may also reduce number to send



## Tests - packet loss rate - with reverse load

- performed on Ethernet

- same procedure as packet loss rate
  send "learn" packets
  for every 4 packets that are received,
    1 is sent back
  test setup did not record stats about return packet

## Tests - back-to-back packets

- definition from RFC 1242:

> **Fixed length frames presented at a rate such that there is the minimum legal separation for a given medium between frames over a short to medium period of time, starting from an idle state.**

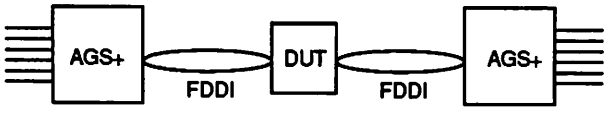
- performed on Ethernet

- procedure:
  - send "learn" packets
  - for each packet size
    - send burst of packets at max rate
    - if all not received
      - decrease length of burst
    - if all received
      - increase length of burst
      - up to max of 1 second's worth
    - find longest burst for all packets passed



## Tests - misc

- alternate FDDI test
  - use Cisco AGS+s with 6 Ethernets as funnels
  - need to have FDDI analyzer to verify rate




- the world is bugs
  - IP 256 byte packet length field
  - IPX length field
  - need more work analyzing test generators
  - modulo - 32



## Reporting the results

- whole lot of data

- throughput
  - lines representing % of theoretical max rate

- packet loss rate
  - graph of % of theoretical in vs % of theoretical out

- back-to-back
  - lines representing % of 1 second's theoretical packets

(archive of results of tests of 10/91 ftp-able from
hsdndev.harvard.edu in pub/rtests/10.91)



## Promo - Harvard Test Lab

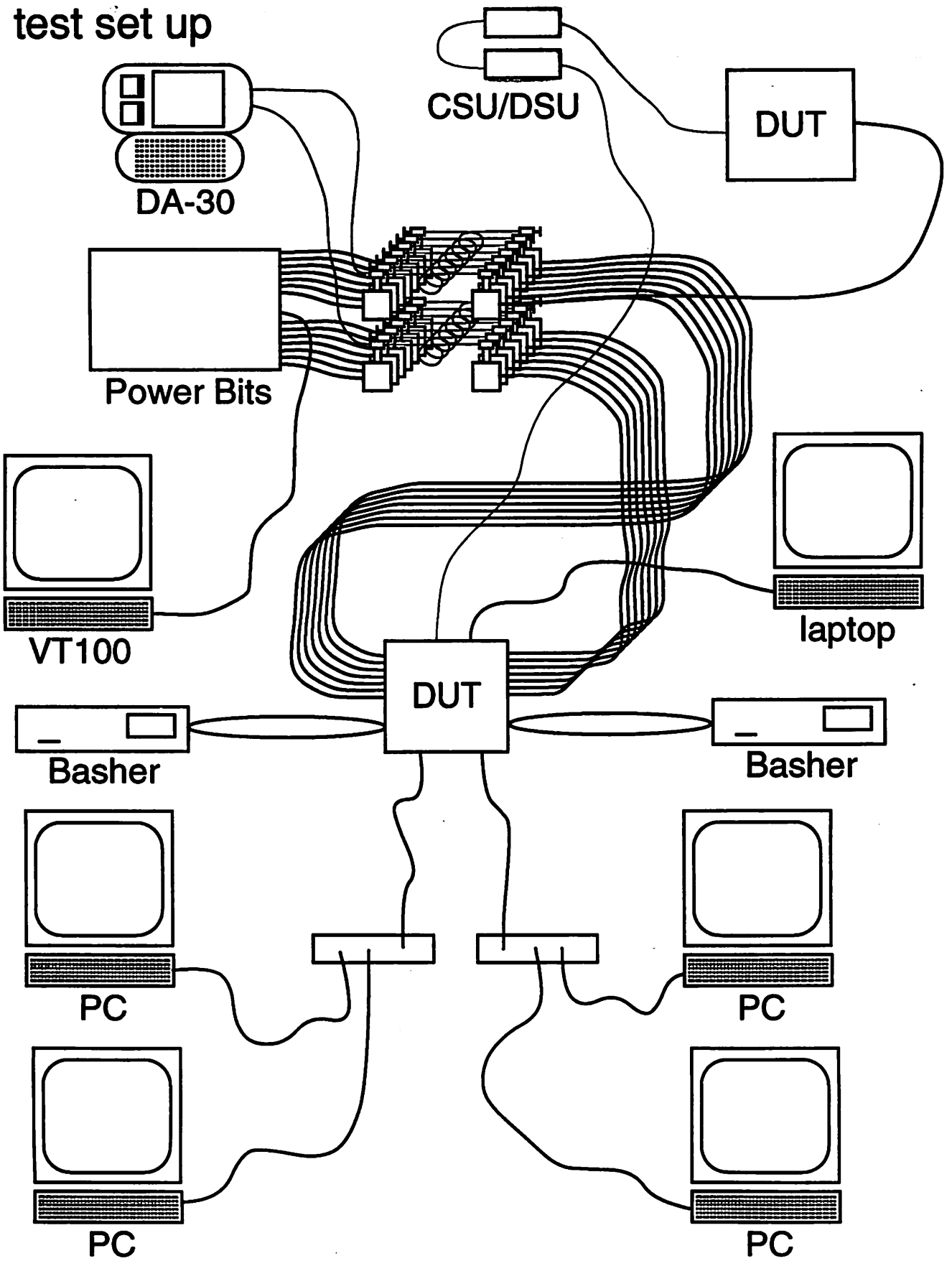
- permanent test lab

- availiable to vendors for private testing
  - development cycle

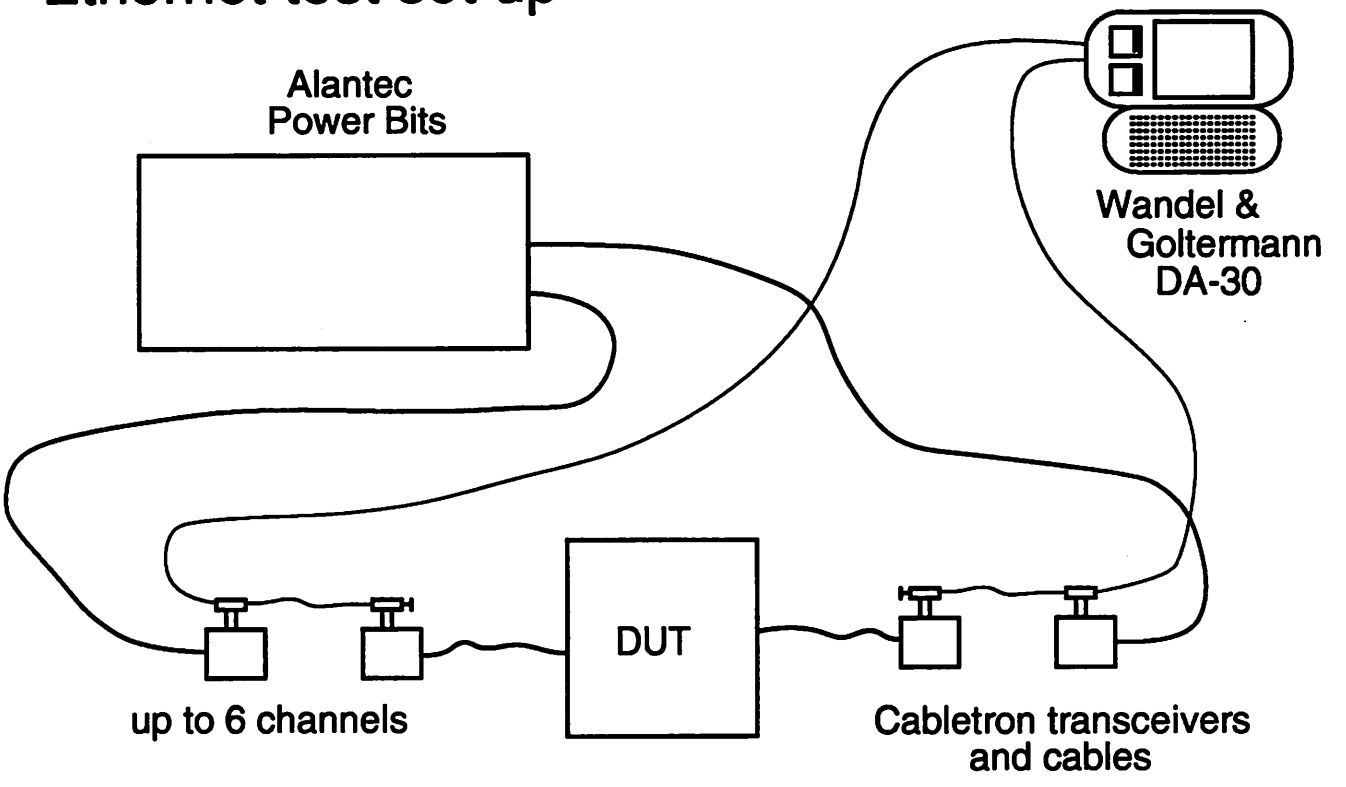
- available to vendors for cirtified testing
  - Harvard certifies that the test was run
  - not pass-fail

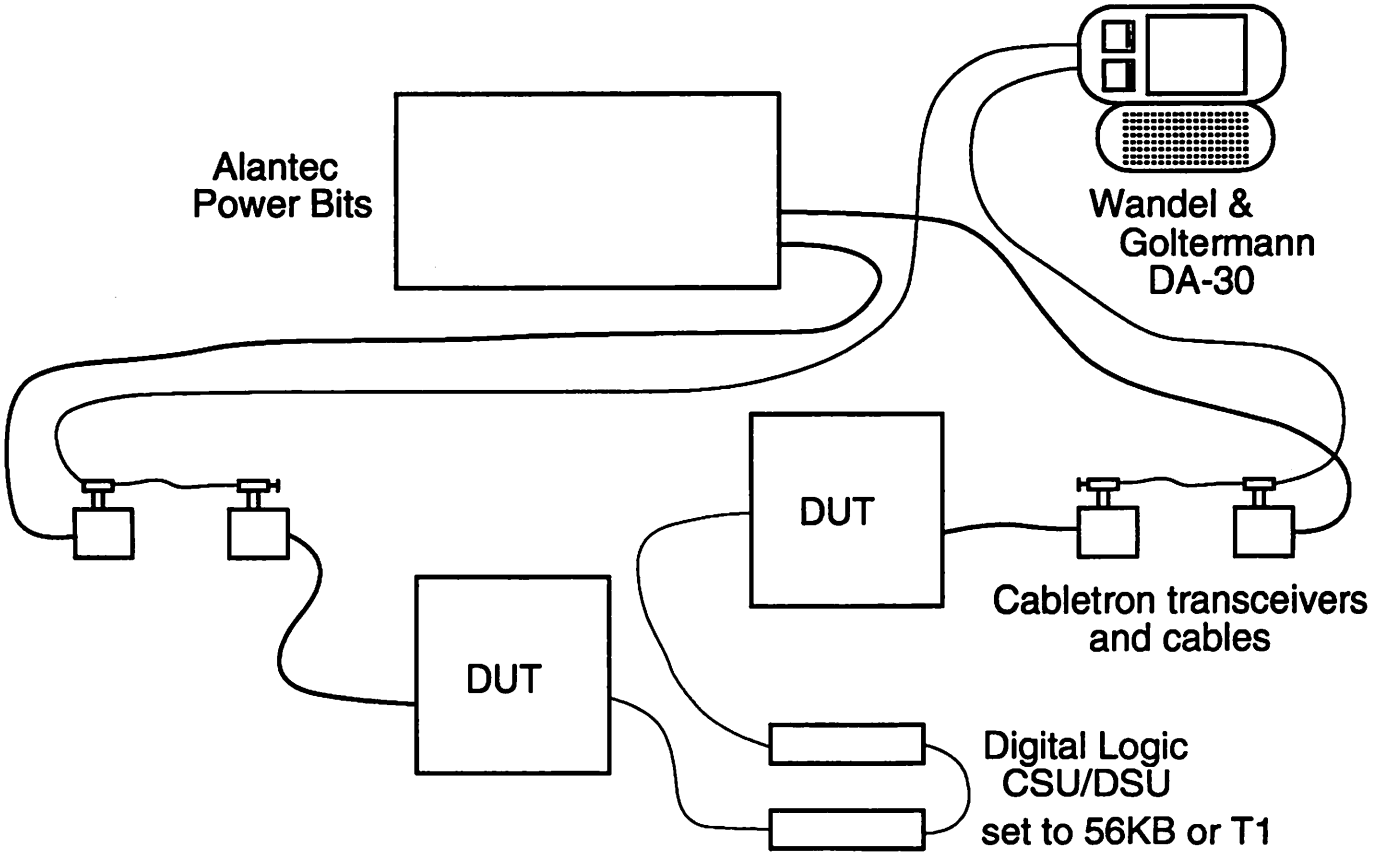
- available to others to test products
  - media
  - potential customers
  - "how do the boxes work in 'my' net"
  - custom protocol mix etc

- cheaper to non-profit organizations

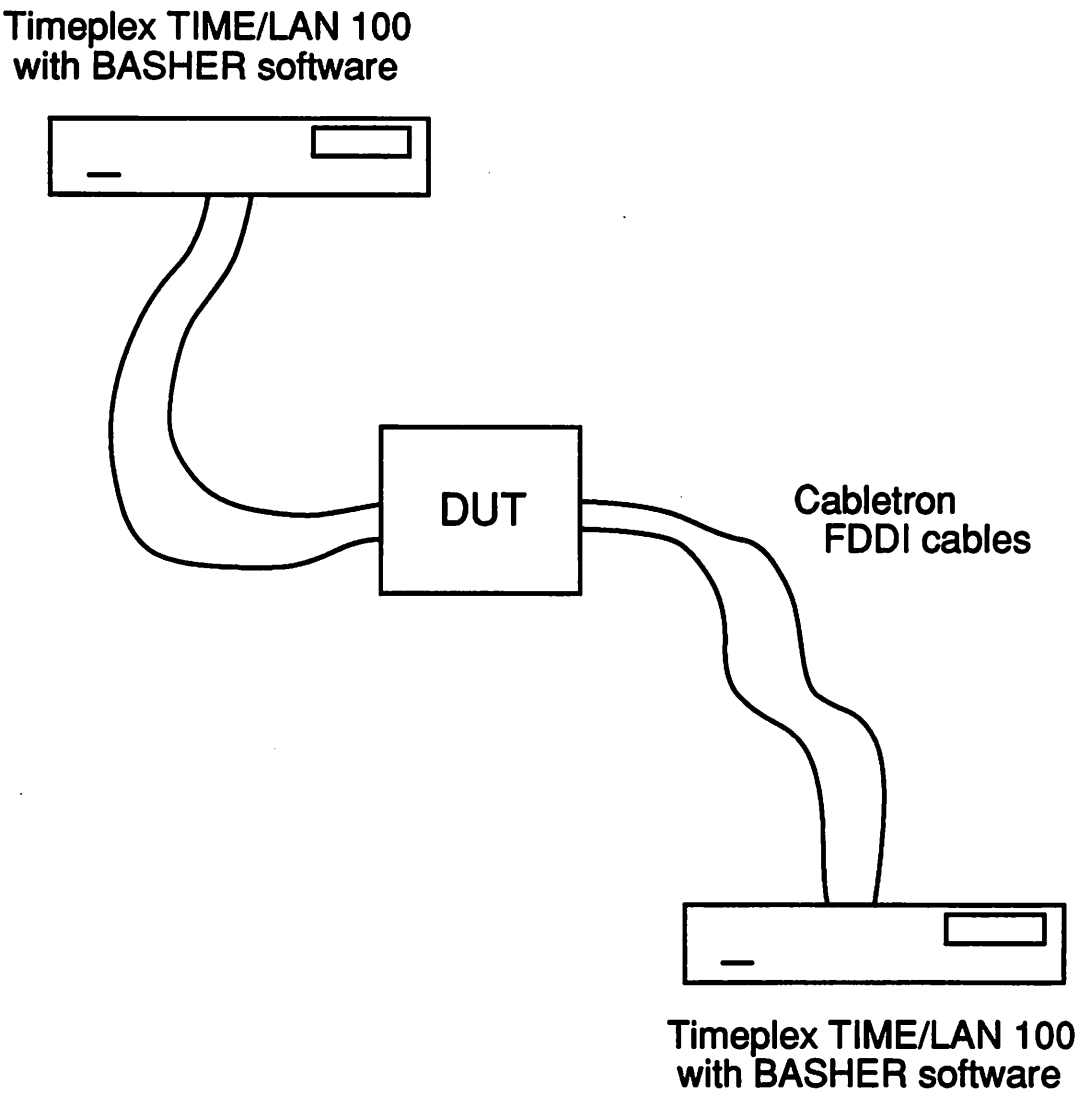

test set up
CSU/DSU
DUT
DA-30
Power Bits
VT100
laptop
DUT
Basher
Basher
PC
PC
PC
PC

# Ethernet test set up

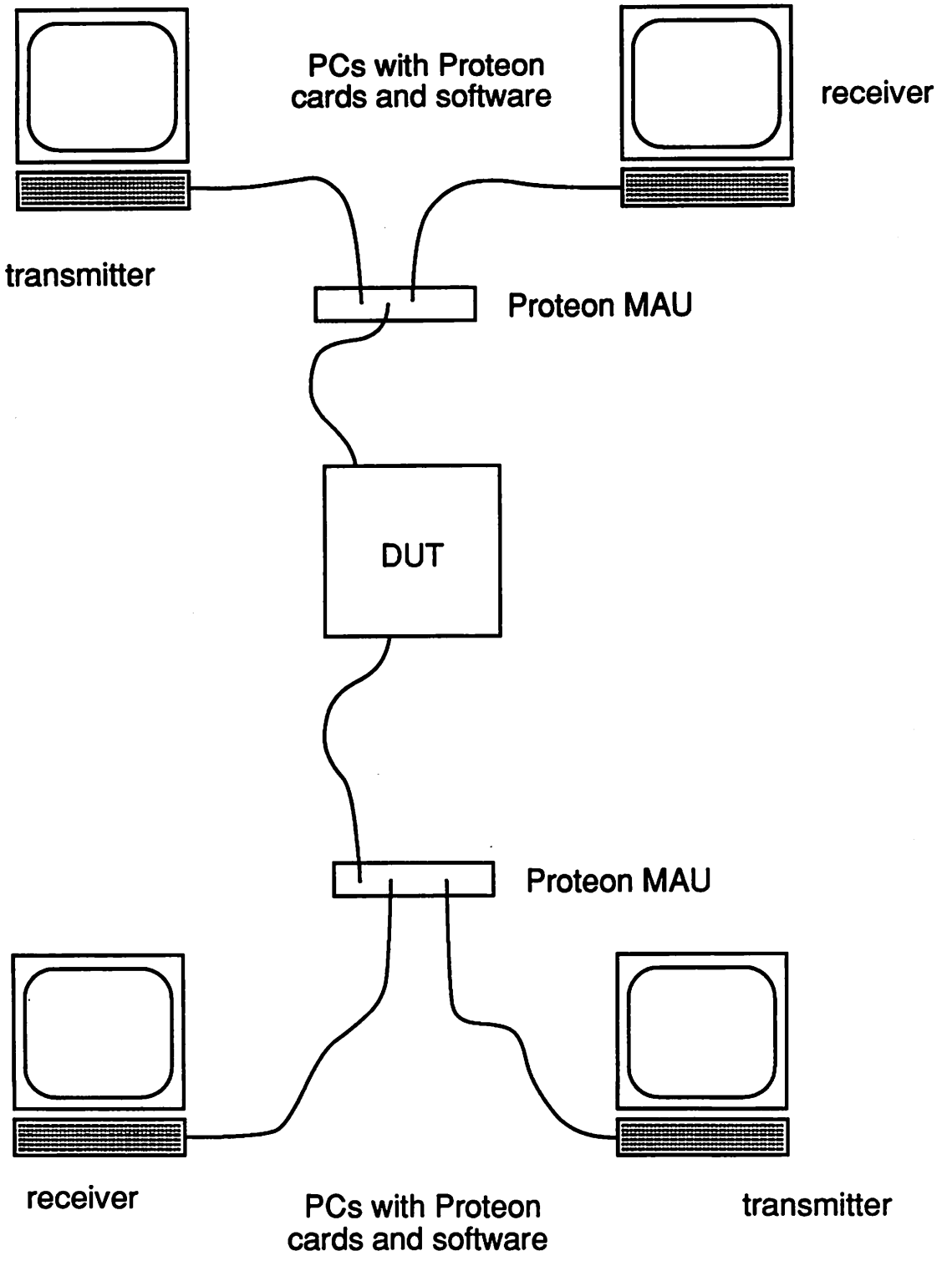


# FDDI test set up



# Token ring test set up



## Ethernet test engine

- built by Alantec - now known as Power Bits
- based on the Alantec Power HUB
- 12 Ethernet ports
- uses script to create and send packets
- some atomic functions
  - stream command
  - burst command
  - measure packet loss rate
  - measure throughput
  - measure back-to-back burst size
- options to commands select:
  - packet size (from list)
  - transmit rate
  - transmit time
  - packet sequence
- tags packets
- designed to be able to implement some of the BMWG tests

(scripts used in 10/91 tests ftp-able from hsdndev.harvard.edu in pub/rtests/10.91)



## Ethernet test system

- Wandel & Golterman - DA30 LAN tester
- 2 Ethernet or token ring ports
- implements same set of tests as Power Bits
- plus latency
- plus LAN analyzer



## FDDI test generator and counter

- Timeplex TimeLAN 100 router with BASHER software
- supports Ethernet, FDDI and token ring
- select:
  - protocol (XNS or IP)
  - packet size (20 to xx bytes)
  - transmit rate ( up to 21,000 pps)
  - transmit count
- puts and checks sequence numbers in packets
- sends 8 packets per token on FDDI



## Token ring test generator and counter

- Proteon special code running on PCs
- select:
  - packet file to send
  - packet size
  - send count
  - transmit rate
    - up to 17,000 pps with 1 packet / token
    - > 20 Kpps with > 1 packet / token

## IP Test packet for Ethernet tests

- IP - UDP Echo Request
  from 192.32.100.1 to 192.32.200.1

```
                        DATAGRAM HEADER

AA 00 04 00 02 04       # dest MAC address (router)
AA 00 04 00 01 04       # src MAC address
08                      # type high byte
00                      # type low byte

                        IP HEADER

45                      # IP version - 4,
                          # header length (4
                          # byte units) - 5
00                      # service field
00 2E                   # total length
00 00                   # ID
40 00                   # flags (3 bits)-4 (do not
                          # fragment),
                          # fragment offset-0
0A                      # TTL
11                      # protocol - 17 (UDP)
C4 8D                   # header checksum
C0 20 64 01             # src IP address
C0 20 C8 01             # dest IP address

                        UDP HEADER

C0 20                   # source port
00 07                   # destination port
                          # 07 = Echo
00 1A                   # UDP message length
00 00                   # UDP checksum

                        UDP DATA

00 01 02 03 04 05 06 07 # some data
08 09 0A 0B 0C 0D 0E 0F
```



## IP ARP Request packet for Ethernet tests

- ARP Request from dest to router

```
                        ARP REQUEST PACKET

FF FF FF FF FF FF
AA 00 04 00 03 04       # from port 6 mac address
08                      # type hi byte
06                      # type low byte
00 01                   # hardware type = ethernet
08 00                   # protocol type = IP
06                      # hw addr length
04                      # proto addr length
00 01                   # opcode = request
36 6 22                 # from port 6 mac address
C0 20 C8 01             # and port 6s IP
FF FF FF FF FF FF       # router's MAC
C0 20 CD 02             # router's IP
```



## Bridge Test packet for Ethernet tests

- Bridge test packet
  UB data - on Ethernet (not 802.3)
  from AA-00-04-00-01-04 to AA-00-04-00-03-04

```
                        DATAGRAM HEADER

AA 00 04 00 03 04       # dest address
AA 00 04 00 01 04       # src address
90                      # type high byte
00                      # type low byte

2C 00                   # length (little endian)
00 00                   # command

                        NON-RANDOM DATA

00 01 02 03 04 05 06 07
```



## DECnet Test packet for Ethernet tests

- DECnet test packet
  short data packet
  from 1.1 to 1.3

```
                        DATAGRAM HEADER

AA 00 04 00 02 04       # dest hardware address
AA 00 04 00 01 04       # source hardware address
60                      # type high byte
03                      # type low byte

                        DATA PACKET

2C 00                   # length (little endian)
81                      # pad?
06                      # flags
00                      # dest area
00                      # dest sub
AA 00 04 00 03 04       # dest ID - tester port 6
00                      # src area
00                      # src sub
AA 00 04 00 01 04       # src ID - tester port 0
00                      # nl2
00                      # visit
00                      # s-class
00                      # pt
```

## DECnet hello packet for Ethernet tests

• DECnet hello from dest to router

```
                        DATAGRAM HEADER

AA 00 00 03 00 00           # dest address, DN multicast
AA 00 04 00 03 04           # src address - tester port 6
60                          # type field - high byte
03                          # type field - low byte

                        HELLO PACKET

22 00                       # packet length (little endian)
0D                          # flags
02 00 00                    # tiver
AA 00 04 00 03 04           # src address - tester port 6
03                          # i info
DA 05                       # blksize (little endian)
01                          # area
00 00 00 00 00 00 00 00     # seed
AA 00 04 00 02 04           # neighbor MAC address
FF 1F                       # timer (litte endian)
FF                          # MPD
02                          # data length
AA AA                       # data
```



## IPX Test packet for Ethernet tests

• Novell IPX test packet

```
                        DATAGRAM HEADER

AA 00 04 00 02 04           # dest MAC is router
AA 00 04 00 01 04           # src MAC address (port 0)
00 2E                       # total length
                                # does not use SSAP
                                # does not use DSAP
                                # does not use control
                                # does not use SNAP

                          IPX HEADER

FF FF                       # checksum - FFFF = none
00 26                       # datagram length
00                          # Transport control - must be 0
02                          # Packet Type - 2 = Echo
00 0D 00 01                 # Dest Network
AA 00 04 00 03 04           # dest node = tester port 6
00 02                       # Dest Socket - 2 = Echo
00 05 00 00                 # Source Network
AA 00 04 00 01 04           # source node = tester port 0
40 02                       # Source Socket

                        ECHO COMMAND

00 01                           # echo request

                          ECHO DATA

00 01 02 03 04 05 06 07
```



## AppleTalk Test packet for Ethernet tests

• AppleTalk DDP Echo Request
  from 5.65 to 107.3

```
                        DATAGRAM HEADER

AA 00 04 00 02 04       # dest MAC address (router)
AA 00 04 00 01 04       # src MAC address
00 2E                   # length field
AA                      # SSAP
AA                      # DSAP
03                      # control
08 00 07 80 9B          # SNAP

                         DDP HEADER

04 26                   # 2 bits 0,
                            # 4 bits hop count,
                            # 10 bits
                            #   datagram length
00 00                   # DDP checksum ( 0= none)
00 6B                   # destination network number
00 05                   # source network number
03                      # dest node ID
41                      # source node ID
04                      # dest socket number
60                      # source socket number
04                      # DDP type - 04 = AEP

                          AEP HEADER

01                      # Echo function
                            # 1 = Echo Request

                          AEP DATA

00 01 02 03 04 05 06 07
```



## AppleTalk AARP Request packet for Ethernet tests

• AARP Request from dest to router

```
                          AARP req

09 00 07 FF FF FF       # dest address (AT broadcast addr)
AA 00 04 00 03 04       # src addr (port 6 mac address)
00 24                   # length
AA                      # SSAP
AA                      # DSAP
03                      # control
00 00 00 80 F3          # SNAP
00 01                   # hardware type (ethernet)
80 9B                   # protocol (AppleTalk)
06                      # MAC length
04                      # protocol address length
00 01                   # request
AA 00 04 00 03 04       # port 6 MAC address
00 00 6B 03             # port 6 AT address (107.3)
00 00 00 00 00 00       # requested MAC address
00 00 6B FD             # router's AT address (107.253)
```

## Test Results: Routers

How fast is fast?



## Thanks

- all test hardware and software provided
  by cooperating vendors

- Alantec
  Power Bits hardware
  programmer time (lots)
- Cabletron
  transceivers and cables
  (esp FDDI cables)
- Cisco Systems
  AGS+ routers with FDDI to act as filters
- Digital Link
  DL551VX extended T1 converter + CSU
- Proteon
  token ring test boards and software
  4 PCs to put them in
- Timeplex
  Time/LAN 100 routers with BLASTER software
- Wandel & Goltermann
  DA-30 Multi-port Dual Protocol Analyzer
  special software
- BMWG
  test design



## Thanks, contd.

- many people at many locations helped produce the
  software and many others helped when at Harvard
  to get things working correctly
  (chronological order)

- Mano Murthy of Alantec
  long hours and fast response
- Hugh Lewis of Timeplex
  got the light to flow
- Earl Veal of HP
  important help in getting things going
- Roger Beeman of Cisco
  careful analysis and wide knowledge critical to
  whatever success in this round
- Nathan Salwen of Proteon
  knows source route packets!
- Norman Magnan of Proteon
  knowledge and enthusiasm
- Bill Krieski of Wandel & Goltermann
  positioning for the future



## oops

- announcement sent on comp.protocols.tcp-ip

- mentioned when sales people called

- some vendors missed
  crunch at end prevented test of 3Com NETBuilder

- a number of requests for ka9q & pcroute
  ran out of time (again)

- not done
  latency
  effects of filters
  effects of broadcast and error packets

- not done enough
  token ring

## Perspective

- 1 GB router on a 9.6KB link not all that useful

- theoretical frame rates for link speeds
  64 octet frames
  "ideal" serial links, i.e., no MAC address
  (rounded up)

| link speed | fps |
|---|---|
| 2.4KB | 5 |
| 9.6KB | 19 |
| 56KB | 110 |
| 64KB | 125 |
| .5T1 | 1508 (*2) |
| T1 | 3016 (*2) |
| Ethernet | 14,880 |
| T3 | >85,000 (*2) |
| FDDI | >400,000 |



## Theoretical rates for Ethernet

single stream

| size | pps | bytes/sec | bits/sec |
|---|---|---|---|
| 64 | 14880 | 892,800 | 7,142,400 |
| 128 | 8445 | 1,047,180 | 8,377,440 |
| 256 | 4528 | 1,141,056 | 9,128,448 |
| 512 | 2349 | 1,191,292 | 9,546,336 |
| 768 | 1586 | 1,211,704 | 9,693,632 |
| 1024 | 1197 | 1,220,940 | 9,767,520 |
| 1280 | 961 | 1,226,236 | 9,809,888 |
| 1518 | 812 | 1,229,368 | 9,834,944 |

(data rate = packet data - FCS)



## Devices - non-routing

- Chipcom
  5102B-EE (bridge module) (software version V1.2)

- Hewlett Packard
  28673a Local Ethernet Bridge
  28674a Remote Ethernet Bridge
  28681a Local Ethernet Bridge

- Kalpana Inc.
  Ether Switch EPS-1500 (no software version)

- Newbridge
  8230 Little Bridge (software version - Release 1)

- RAD Network Devices
  LEB-1 (software version 0.00 Beta)
  REB-10 (software version 3.02)

- Synernetics Inc.
  Lanplex 5000 (software version 1.0.0)



## Devices - routing

- Alantec
  Power Hub (software version 1.0)

- BBN Communications
  T/20 Internet Router (software version 1.2)

- Chipcom
  5102R-EE Router Module (software version 8.2(5))

- Cisco Systems, Inc.
  IGS (software version 8.2(5))
  AGS+ (software version 8.3)

- Hewlett Packard
  27285a Local Router

- Network Systems Corporation
  6800 (software version 3.0)

- Proteon
  CNX500 (software version 11.0)

- SUN Microsystems
  SS2

- Timeplex
  TIME/LAN 100 Router*Bridge (software version 2.1.1)

## Results - Ethernet throughput

- still getting faster
  - more "wire speed" routers
    - (for some frame sizes)
- 3 devices > 14,500 pps (64 byte)
- more > 10Kpps

- cluster of mid range devices
  - ~6K FPS @ 64 bytes

- 14 devices > 99% at 1518



## Results - Ethernet multi stream

- hard to do good multi-stream test, hard to see actual peak rates

- only did packet loss rate
  - don't know what throughput would mean

- useful in cases where LANs converge

- with 6 streams (input 89280 pps @ 64 bytes)
  - 4 devices above 60,000 pps aggregate
  - 2 routers, 2 non-routers

- with 6 streams (input 59 M bits/sec @ 1518 bytes)
  - 3 devices > 99%
    - one perfect, another lost 1 packet in 30 sec
  - 1 other device > 98%



## Results - Ethernet back-to-back

- good test of queue size

- a number of devices > 1 second at 1518 byte



## Results - FDDI

- early in the FDDI game

- how deal with >1 packet/token

- GAS+s as funnels seems like a good tack but need to calibrate

- 1 device > 40% of FDDI pipe @ 1518 bytes

- may have run into tester limit
  - too many devices had about the same limits

## Results - token ring

- in development phase of tester software
- hope to use the W&G DA-30



## ests - Ethernet

- fastest - small packet: (alpha order)
  Alantec, Cisco, Kalpana, Synernetics
- fastest - large packets:
  Alantec, BBN, 2 chipcom devices, Cisco ags, Kalpana, 4 HP devices, NSC, Newbridge, Proteon, Timeplex, Synernetics
- highest aggregate
  Alantek, Cisco, Kalpana, Synernetics
- fastest on WAN
  all that had WAN interfaces
- best pkg:
  HP
- most power (also best positioned)
  NSC (7 RISC processors in router tested)
- most suprising
  Newbridge 8230 "Little Bridge"



## ests - FDDI

- fastest using Timeplex tester
  small packets: Cisco, Proteon, Timeplex
  large packets: Cisco, NSC, Proteon, Timeplex
- fastest using AGS+ funnels
  Cisco (Proteon very good but hung)



## ests - token ring (16MB)

- only two devices fully tested
- fastest
  Proteon

## this time and next time

this time:

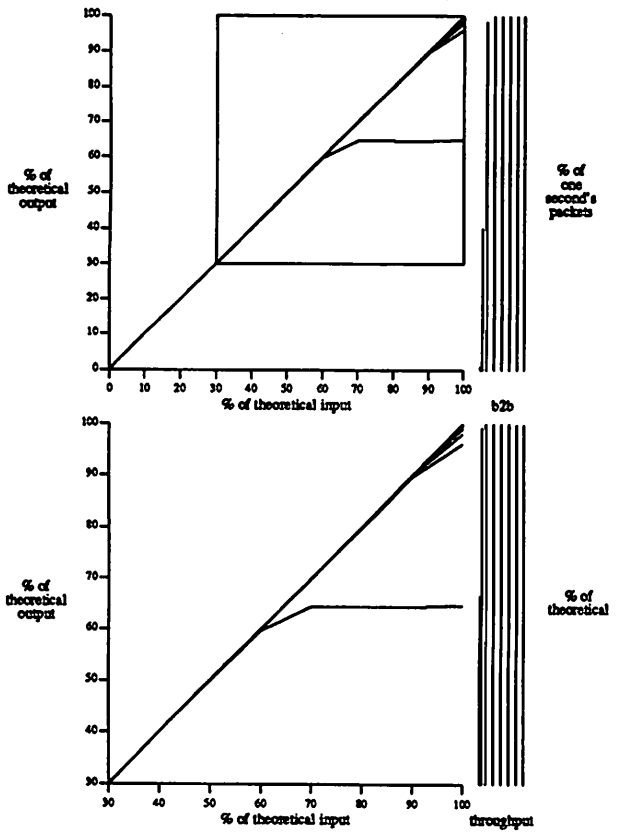
- Power Bits tester fully automatic
  - very easy - start and go away
  - needs some features
    - data file names etc
- DA-30
  - also fully automated

next time

- get tester software done LONG time before
- automate FDDI & token ring tests
- longer test period
- suggestions welcomed

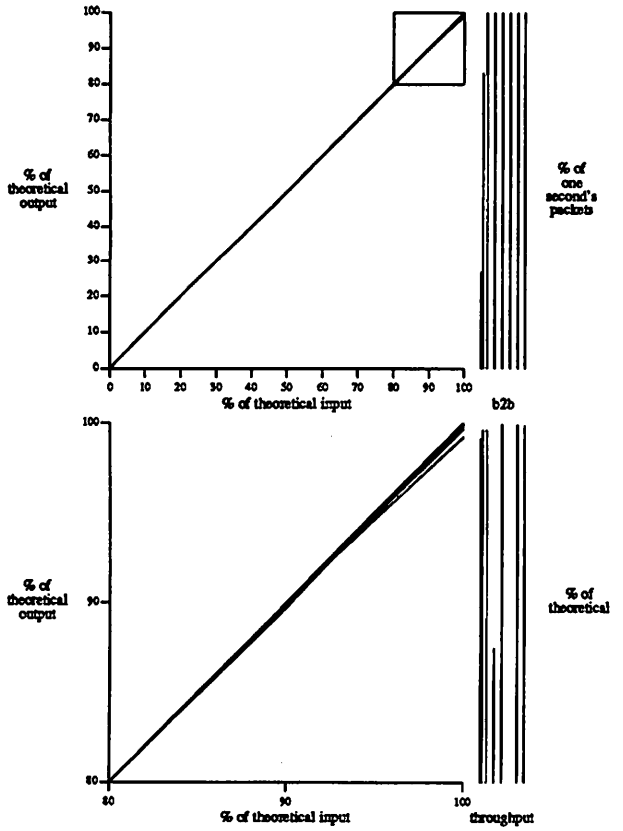
Vendor: Chipcom
Protocol: Bridge

Product: 5102B-FE
organization: within



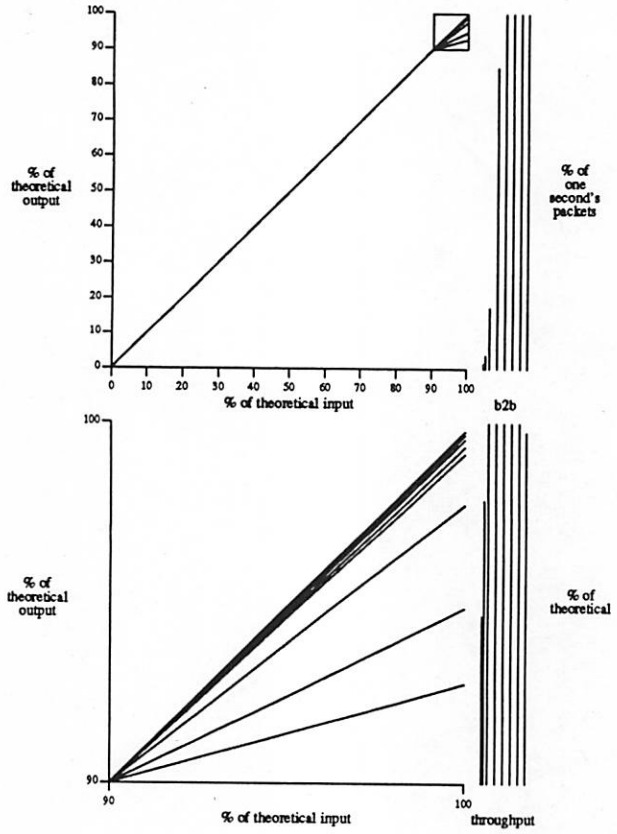
Vendor: HP
Protocol: Bridge

Product: 28673a
organization: within



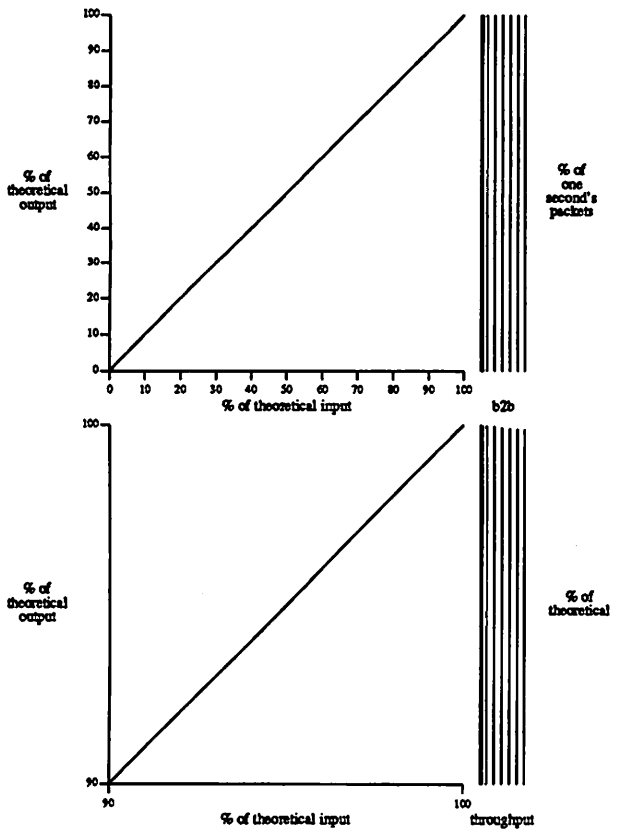
Vendor: HP
Protocol: Bridge

Product: 28681a
organization: within



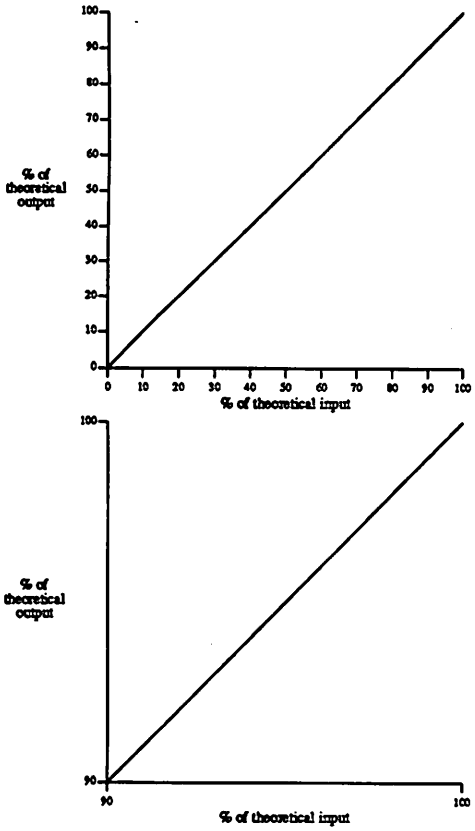
Vendor: Kalpana
Protocol: Bridge

Product: EtherSwitch
organization: between



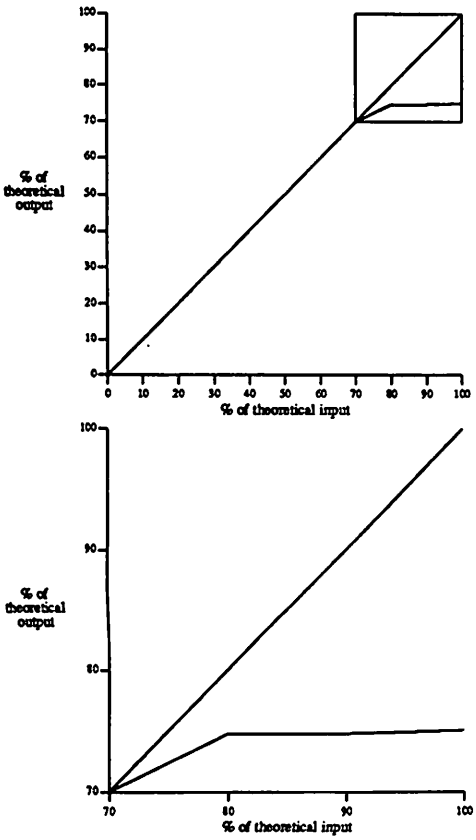
Vendor: Kalpana
Protocol: Bridge

Product: EtherSwitch
Streams: 4



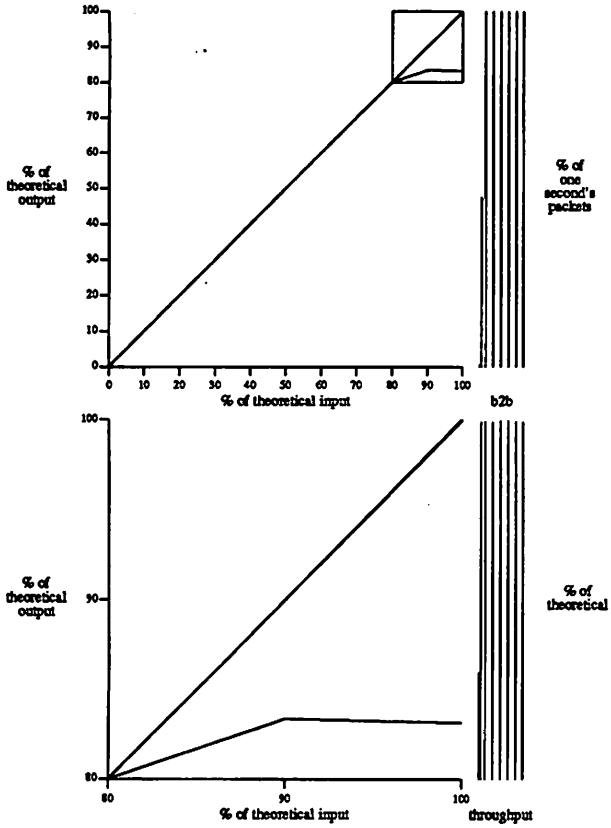

Vendor: Kalpana
Protocol: Bridge

Product: EtherSwitch
Streams: 6



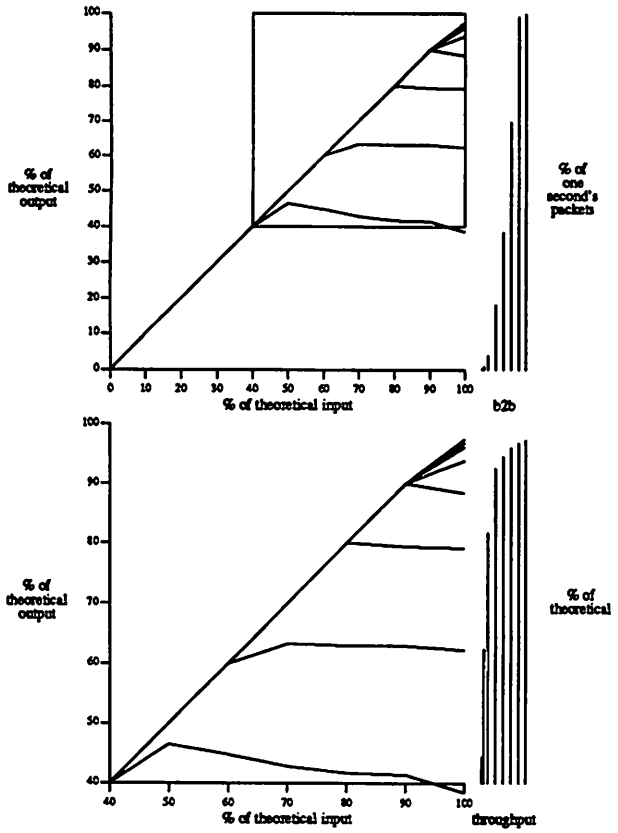
Vendor: Newbridge
Protocol: Bridge

Product: 8230_LittleBridge
organization: within



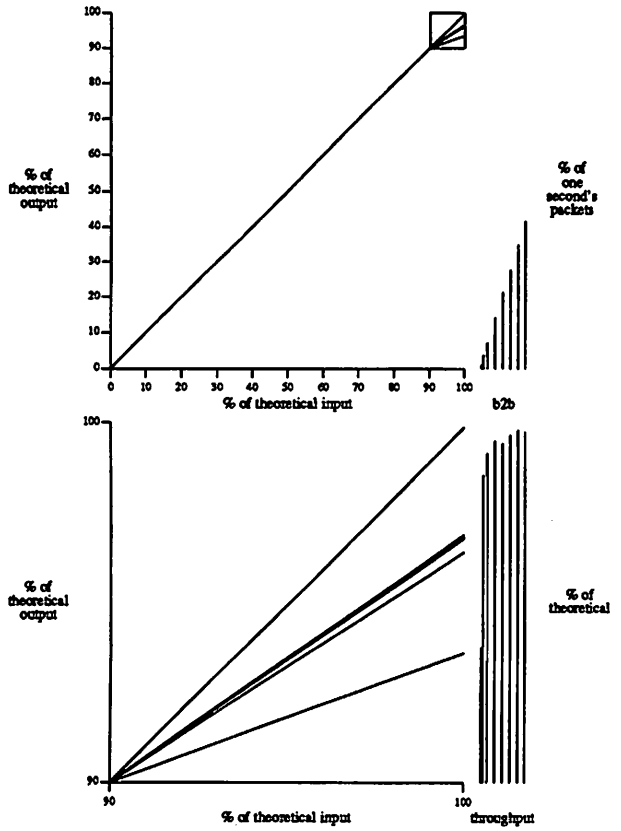
Vendor: RAD
Protocol: Bridge

Product: LEB-1
organization: within



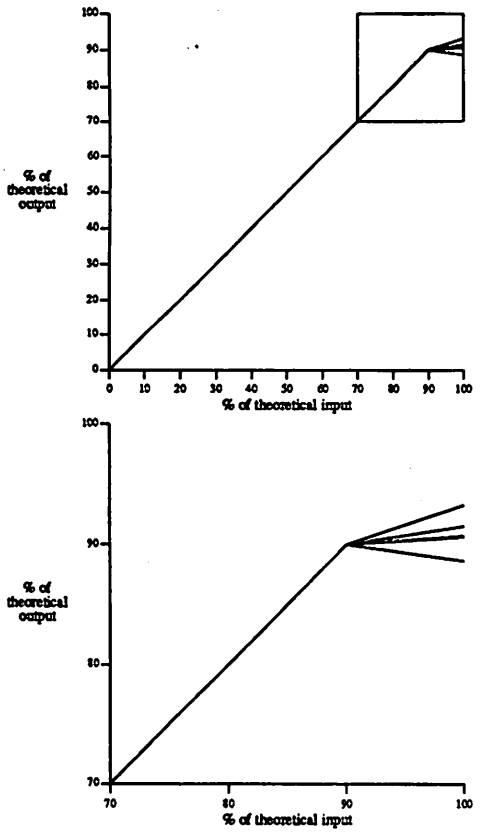
Vendor: Synernetics
Protocol: Bridge

Product: Lanplex
organization: between



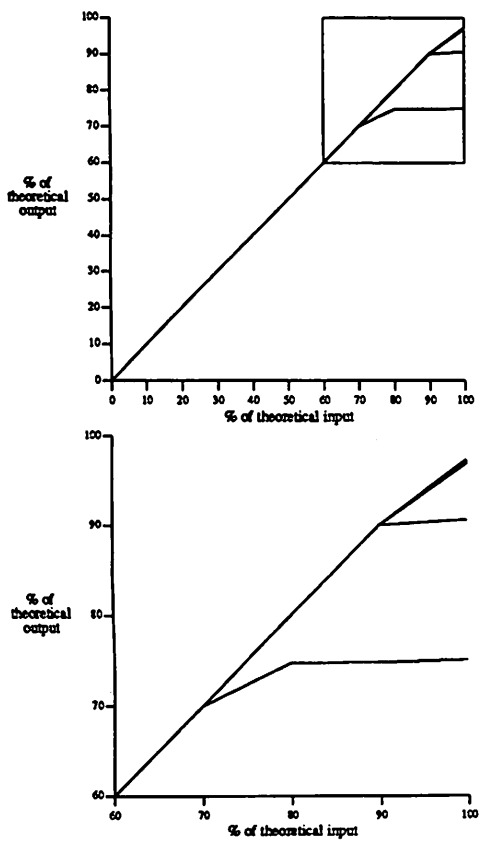
Vendor: Synernetics
Protocol: Bridge

Product: Lanplex
Streams: 4



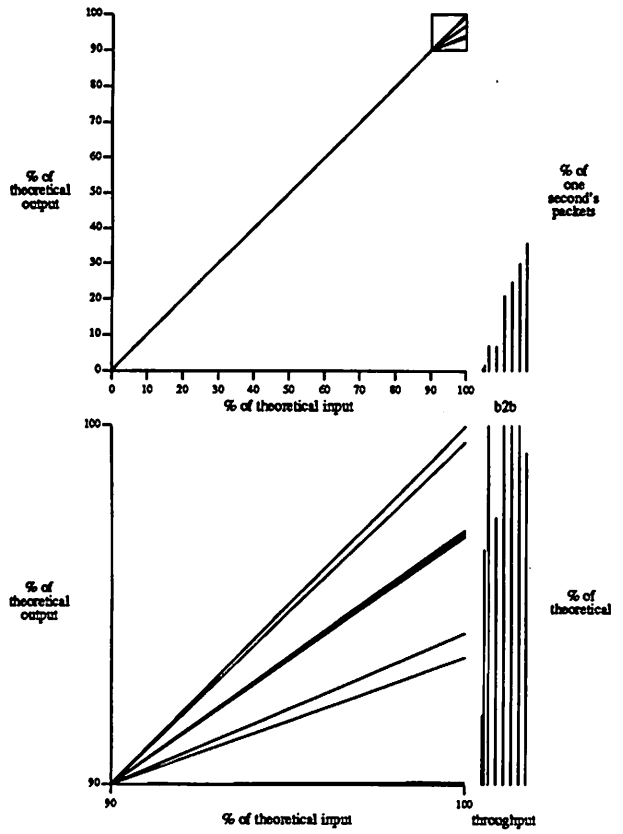

Vendor: Synernetics
Protocol: Bridge

Product: Lanplex
Streams: 6



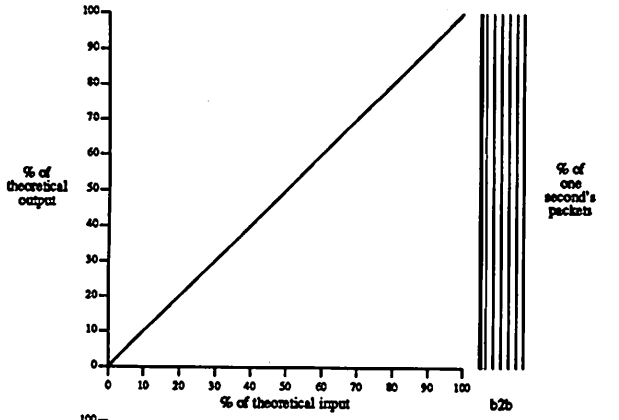
Vendor: Synernetics
Protocol: Bridge

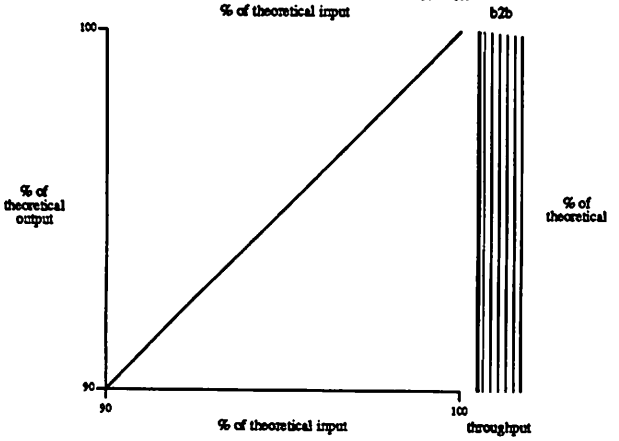
Product: Lanplex
organization: within



Vendor: none
Protocol: Bridge

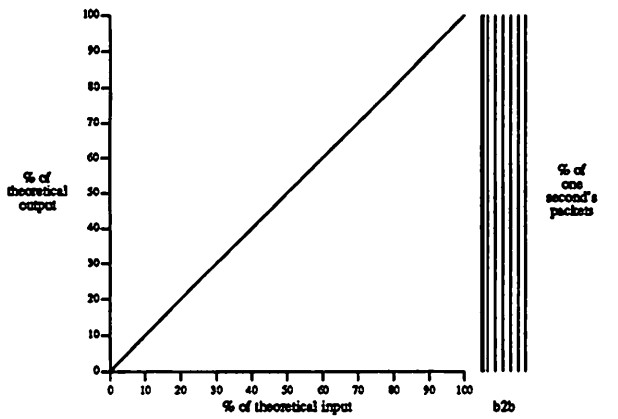
Product: wire
organization: within

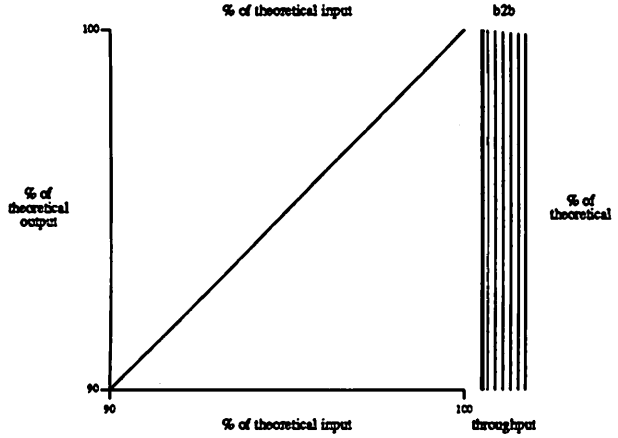


Vendor: Alantec
Protocol: Bridge

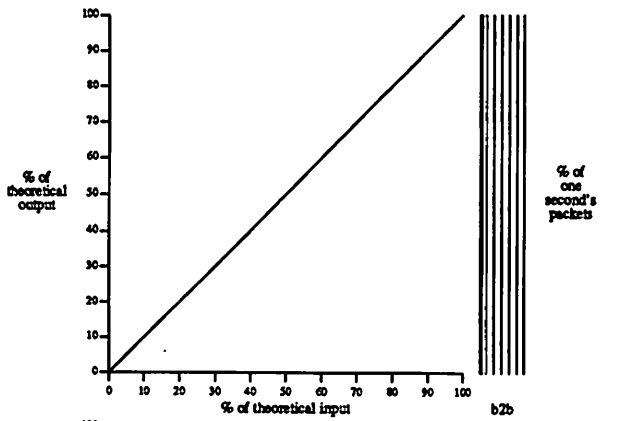
Product: Power
organization: between

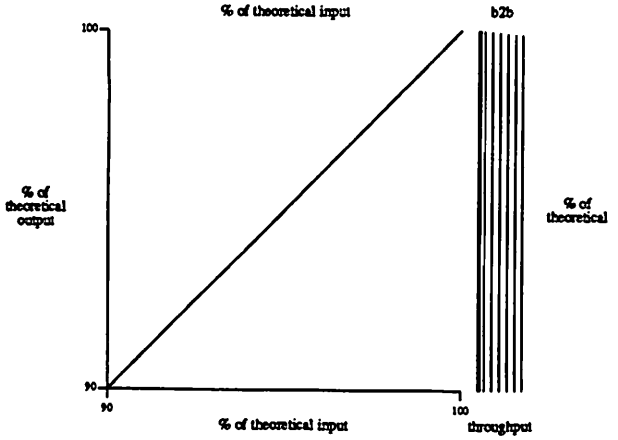



Vendor: Alantec
Protocol: TCP/IP

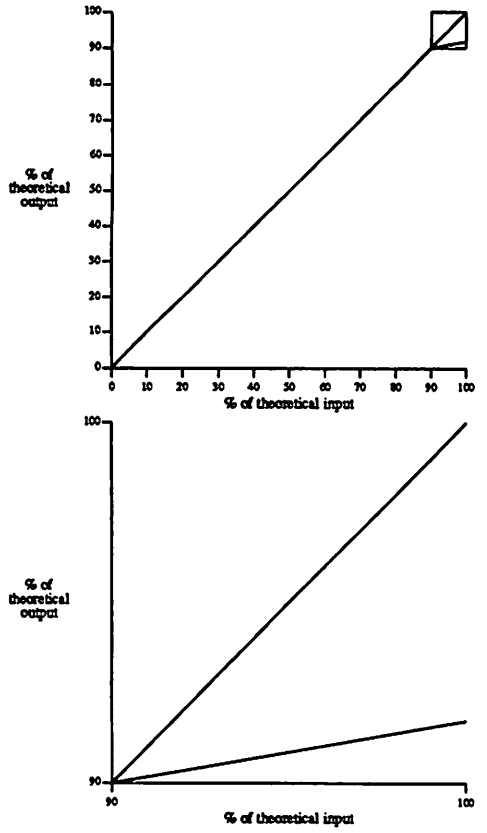
Product: Power
organization: between



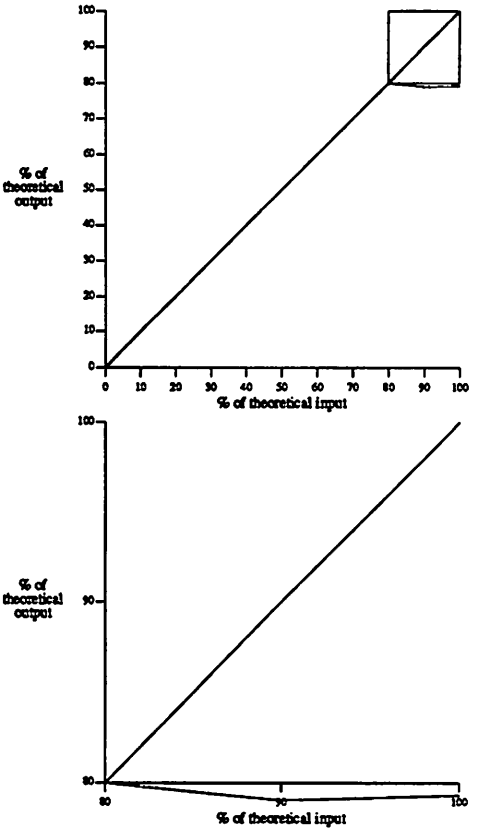
Vendor: Alantec
Protocol: Bridge

Product: Power
Streams: 4



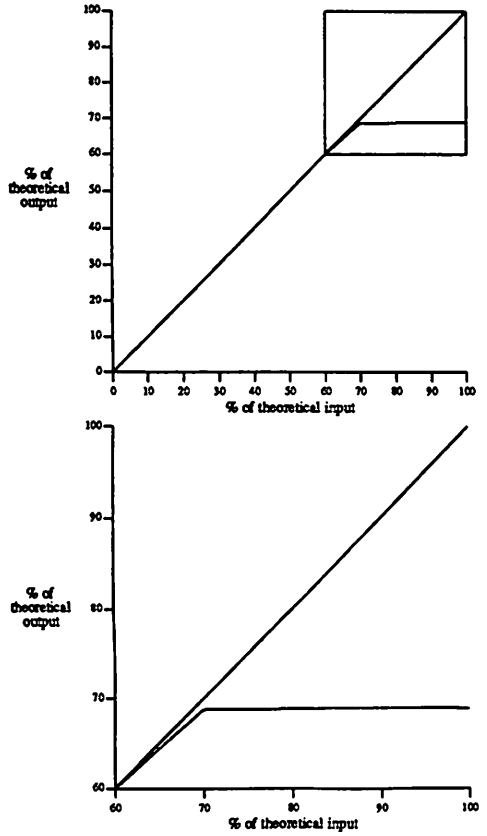

Vendor: Alantec
Protocol: TCP/IP

Product: Power
Streams: 4



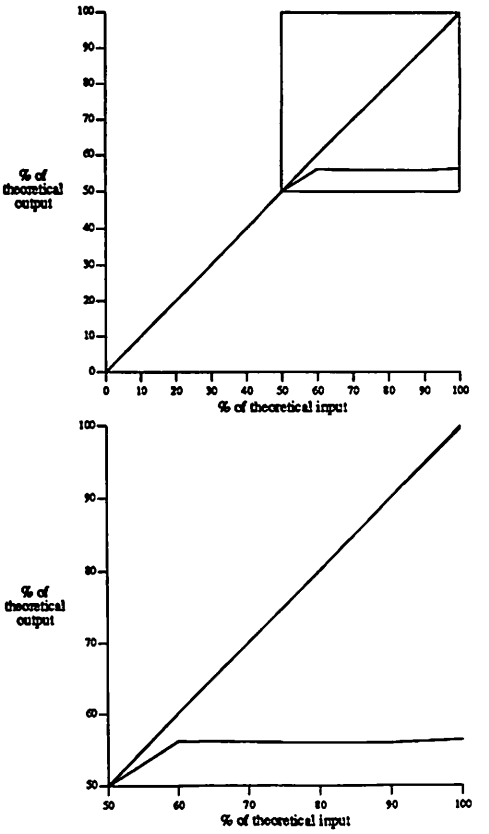

Vendor: Alantec
Protocol: Bridge

Product: Power
Streams: 6



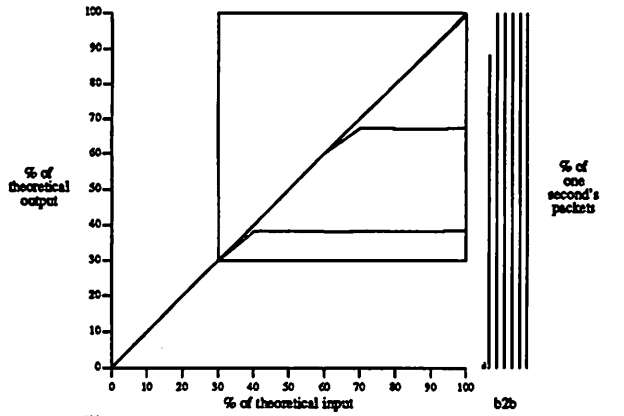

Vendor: Alantec
Protocol: TCP/IP

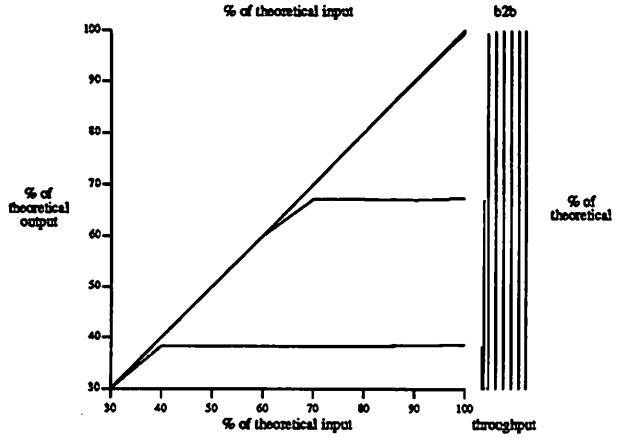
Product: Power
Streams: 6



Vendor: BBN
Protocol: TCP/IP

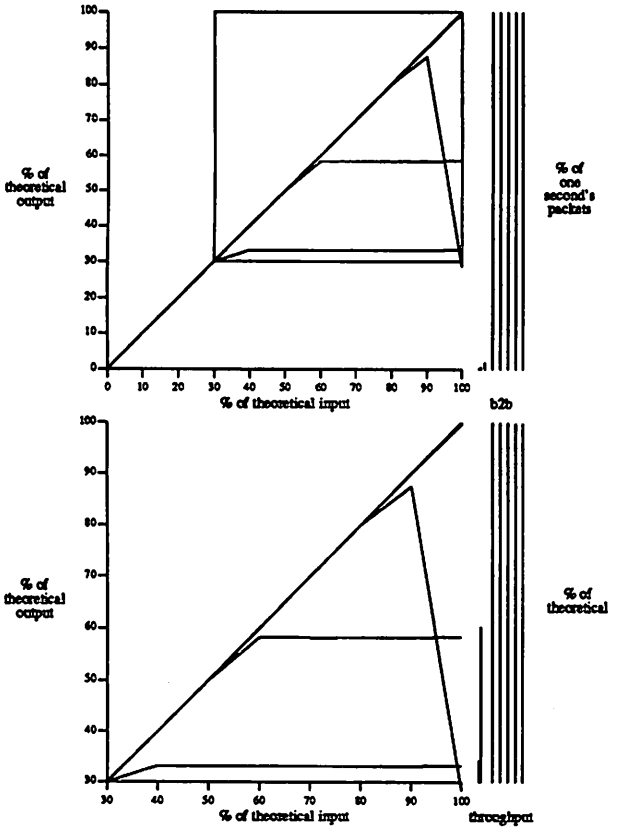
Product: T/20
organization: between



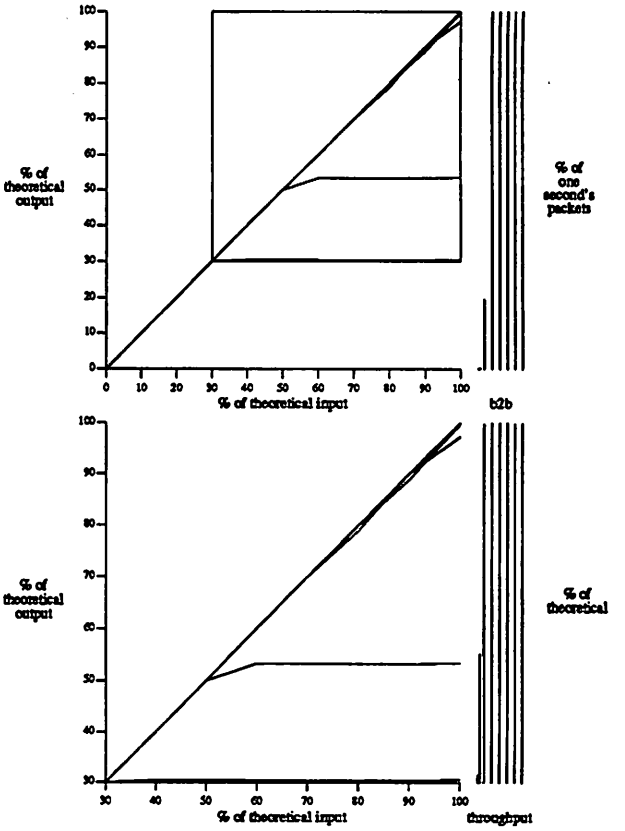
Vendor: Chipcom
Protocol: Bridge

Product: 5102R-EE
organization: within



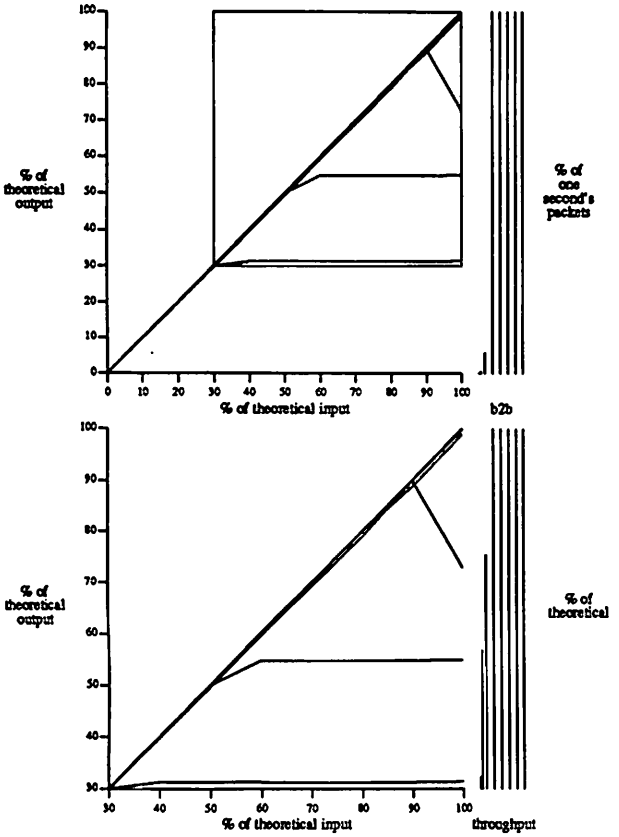

Vendor: Chipcom
Protocol: IPX

Product: 5102R-EE
organization: within



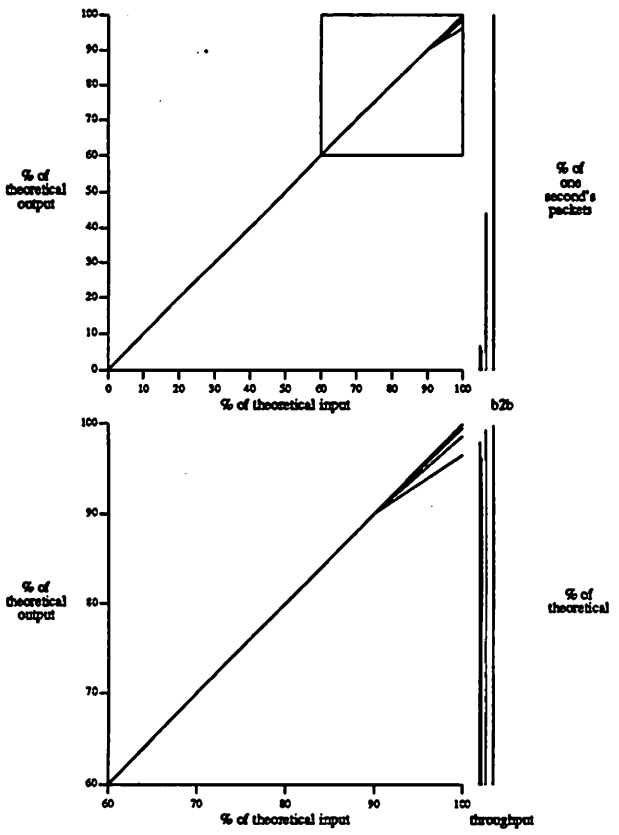

Vendor: Chipcom
Protocol: TCP/IP

Product: 5102R-EE
organization: within



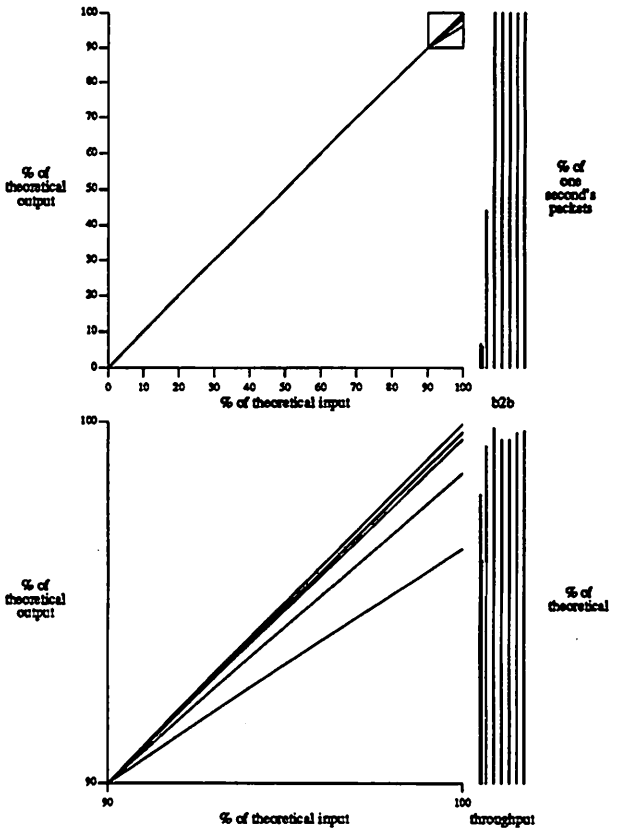
Vendor: Cisco
Protocol: AppleTalk II

Product: AGS+
organization: between



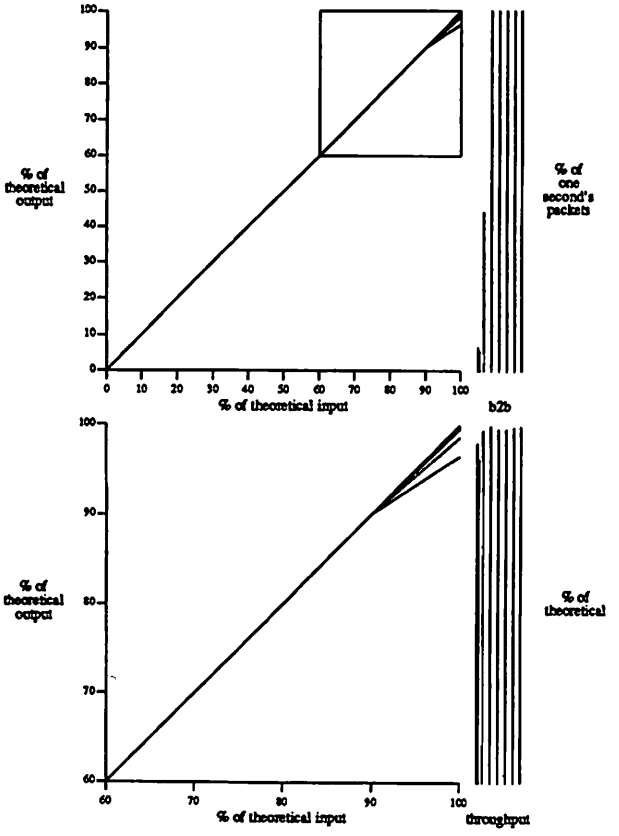

Vendor: Cisco
Protocol: TCP/IP

Product: AGS+
organization: between



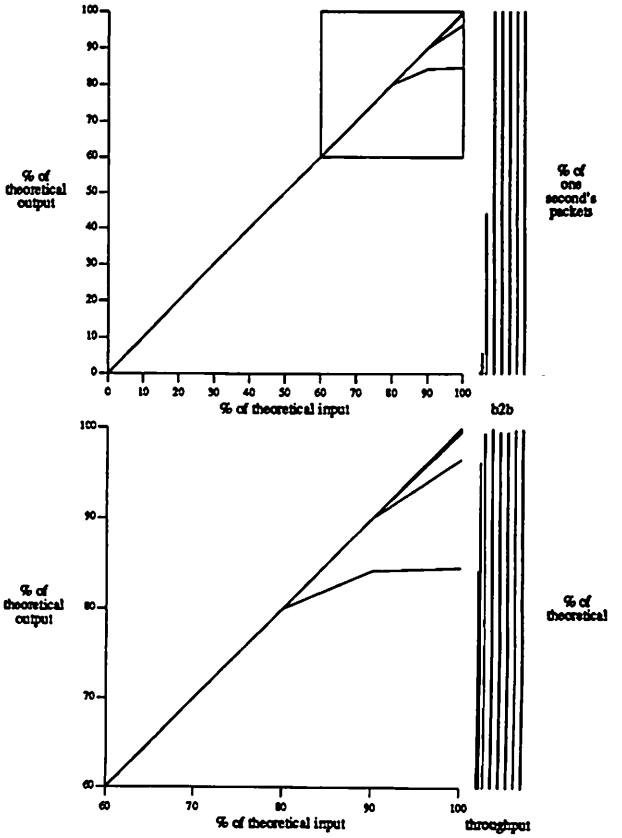

Vendor: Cisco
Protocol: Bridge

Product: AGS+
organization: between



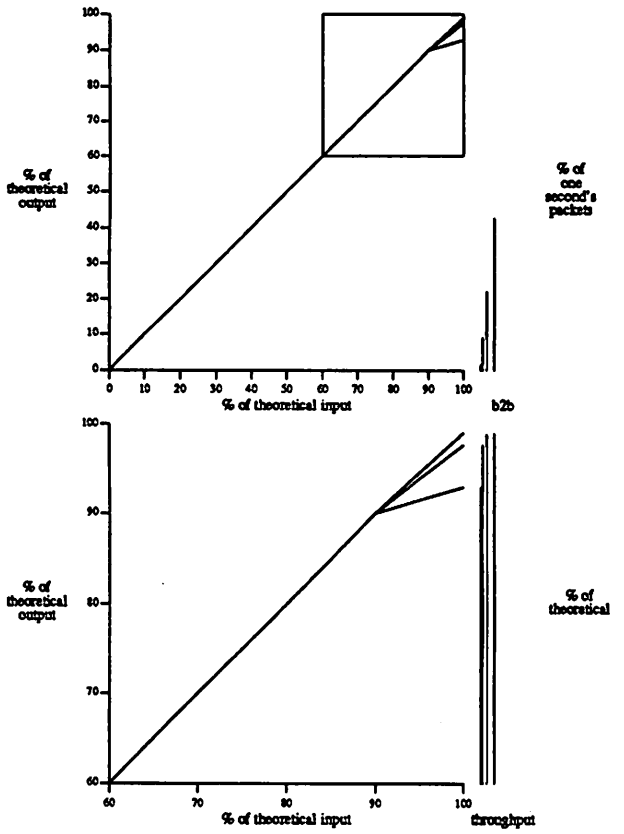

Vendor: Cisco
Protocol: IPX

Product: AGS+
organization: between



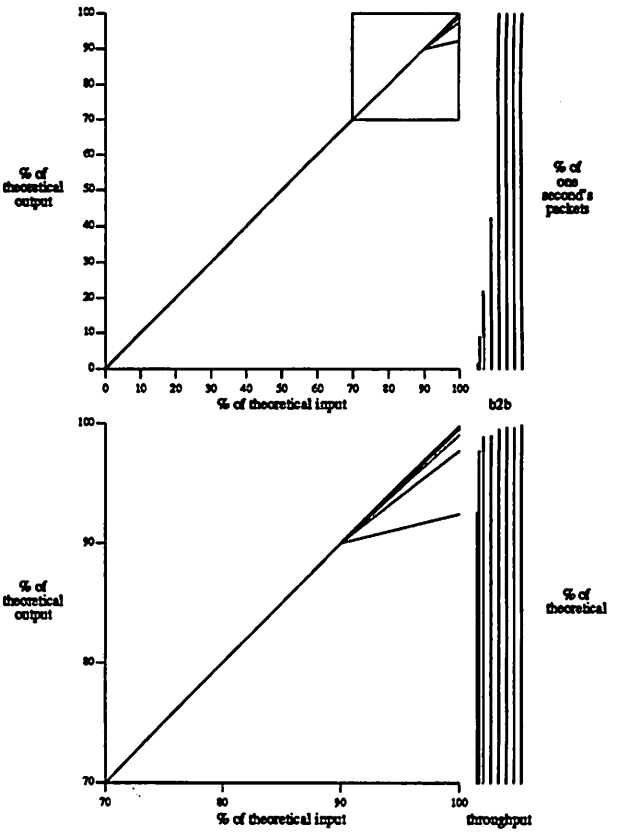
Vendor: Cisco
Protocol: AppleTalk II

Product: AGS+
organization: within



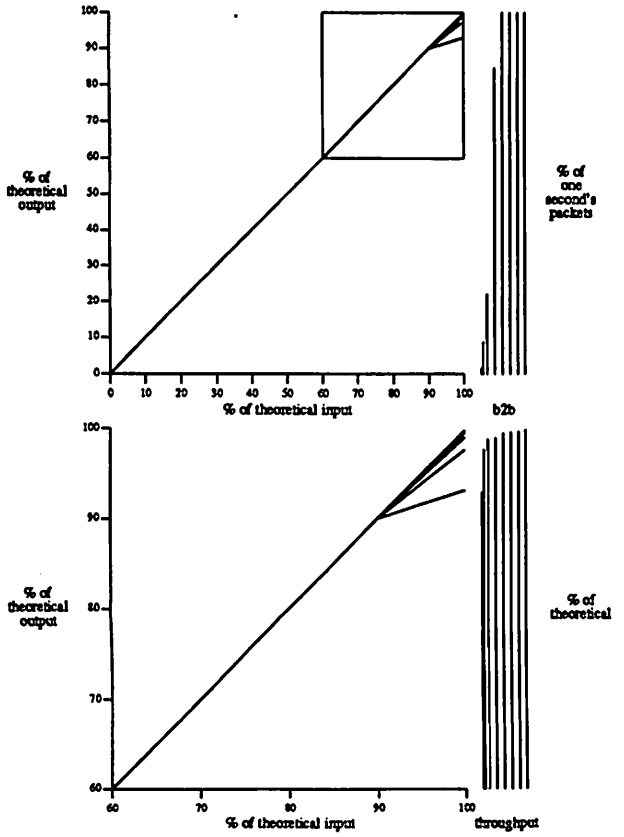

Vendor: Cisco
Protocol: TCP/IP

Product: AGS+
organization: within



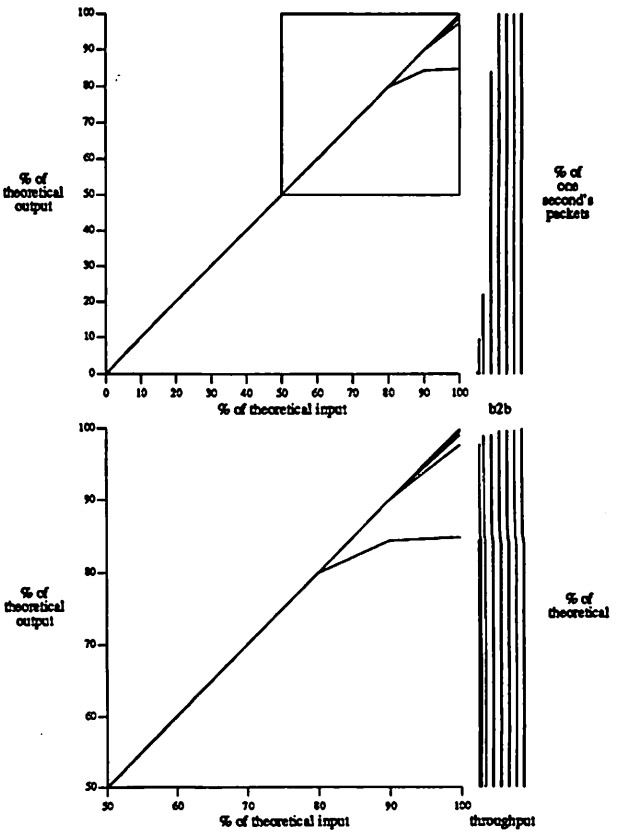

Vendor: Cisco
Protocol: Bridge

Product: AGS+
organization: within



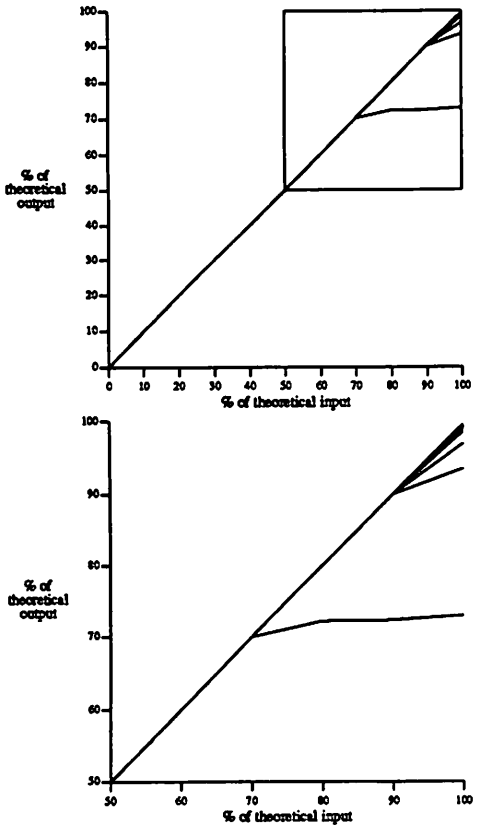

Vendor: Cisco
Protocol: IPX

Product: AGS+
organization: within



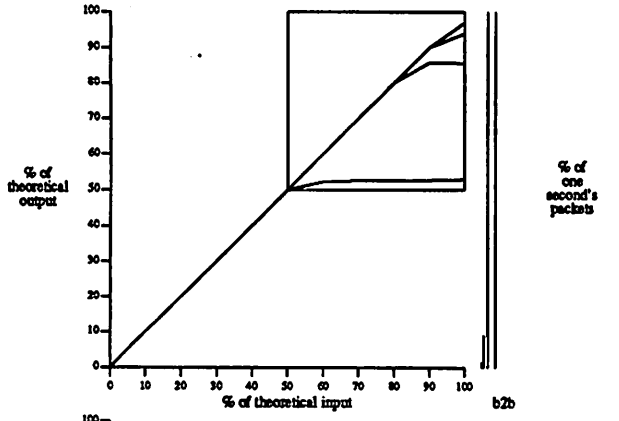
Vendor: Cisco
Protocol: TCP/IP

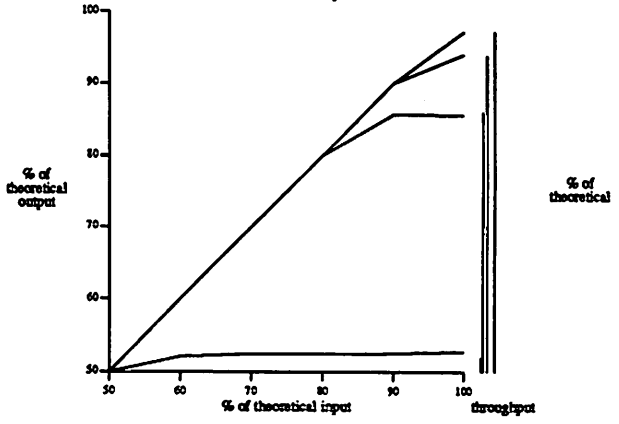
Product: AGS+
Streams: 6



Vendor: HP
Protocol: AppleTalk II

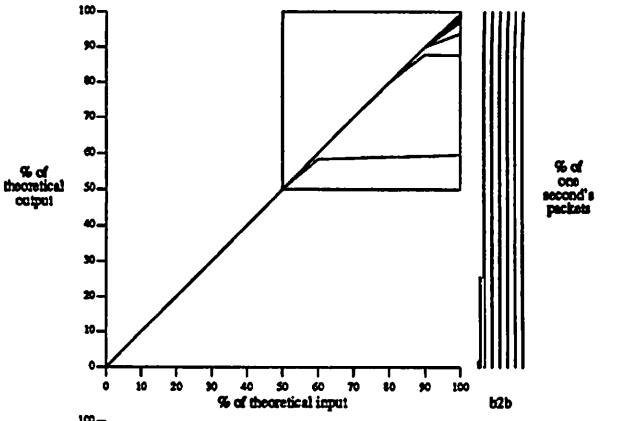
Product: 27285a
organization: within

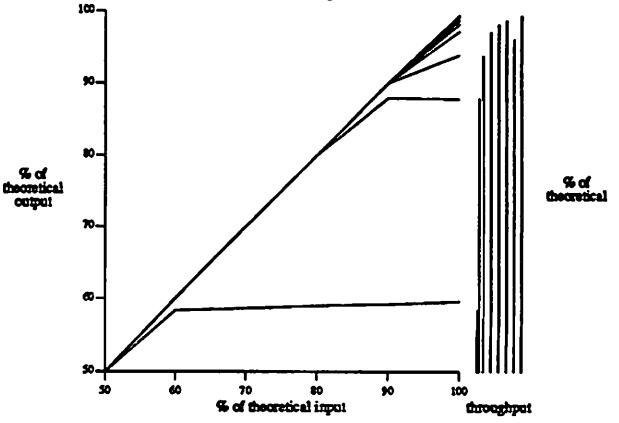



Vendor: HP
Protocol: TCP/IP

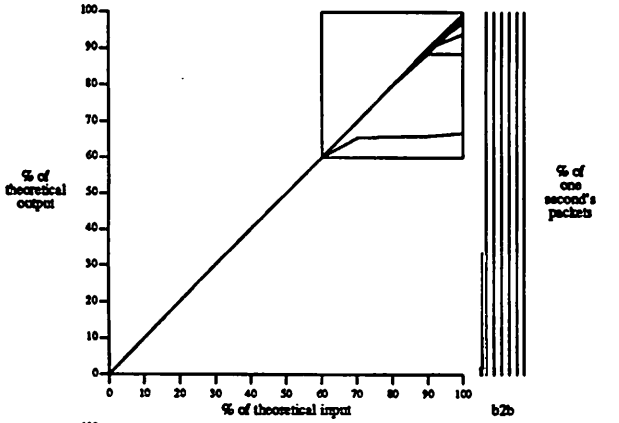
Product: 27285a
organization: within

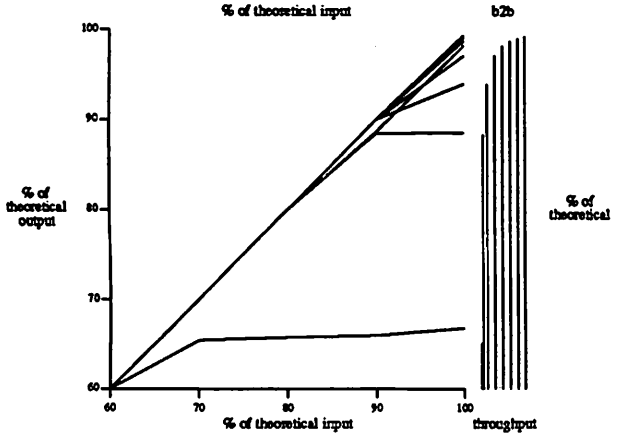



Vendor: HP
Protocol: Bridge

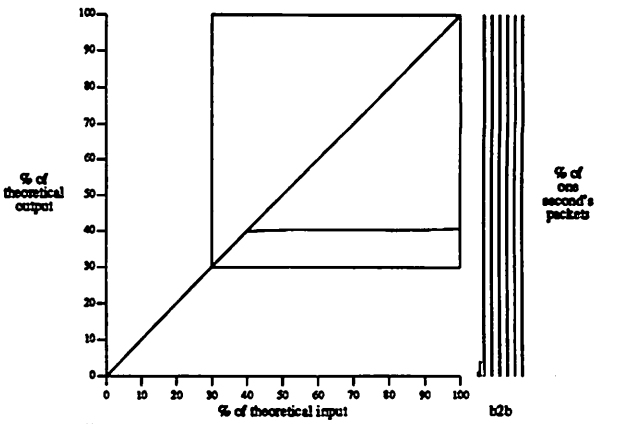
Product: 27285a
organization: within

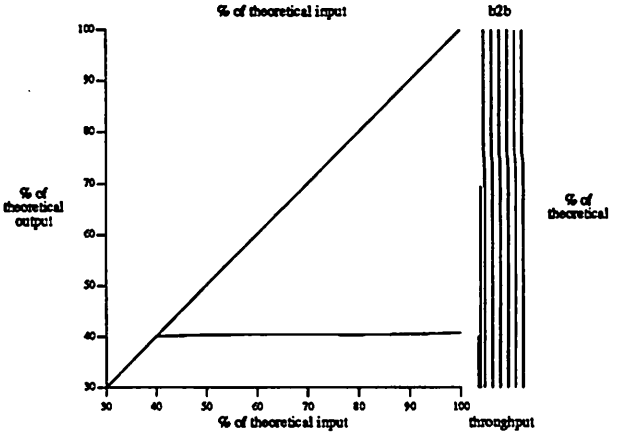



Vendor: HP
Protocol: IPX

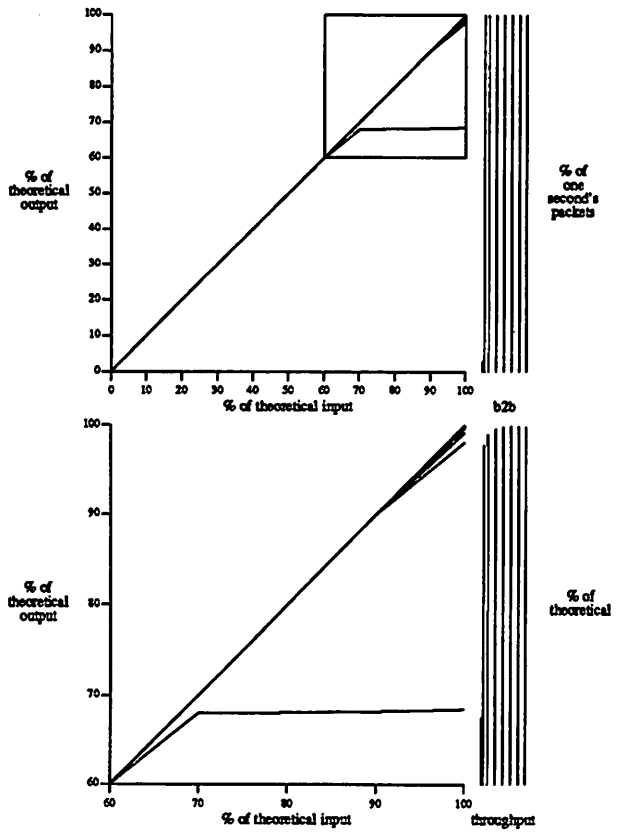
Product: 27285a
organization: within



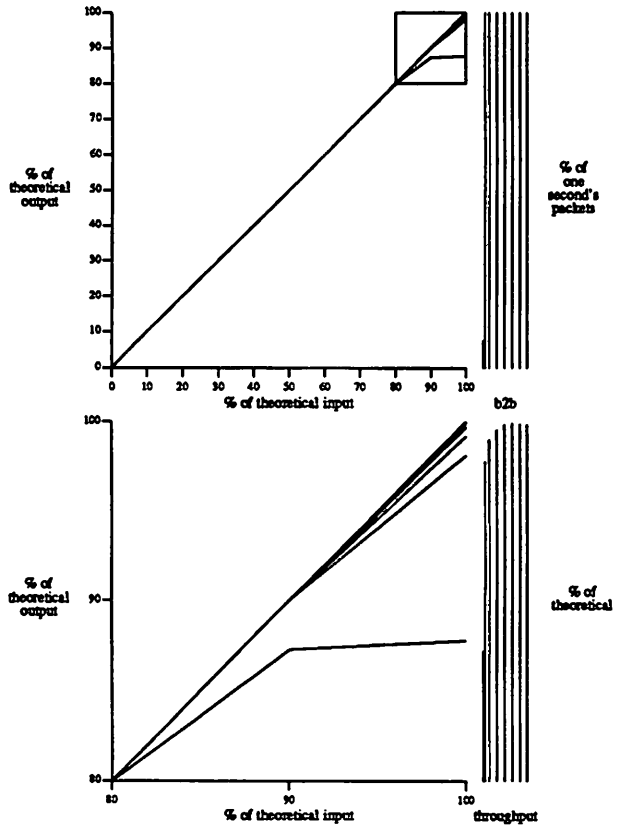
Vendor: NSC
Protocol: TCP/IP

Product: 6800
organization: between



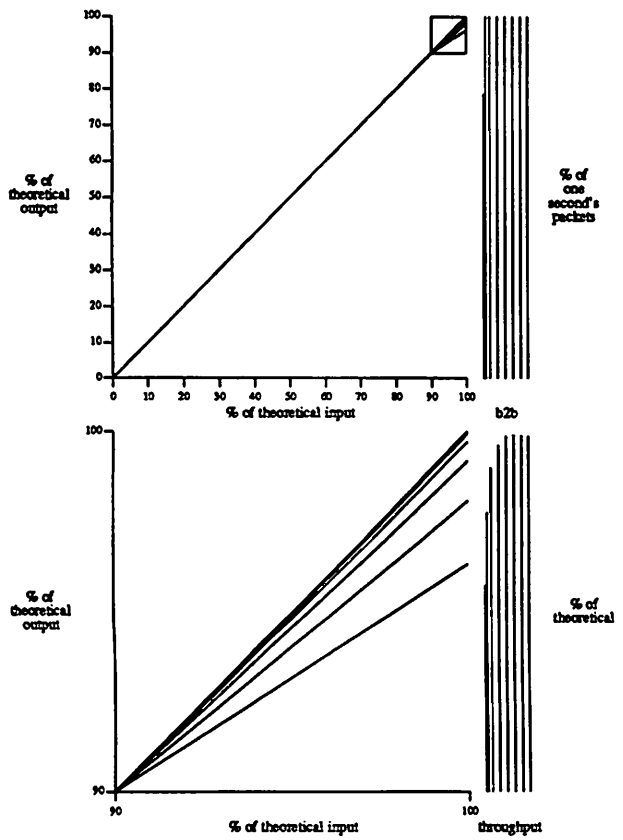
Vendor: NSC
Protocol: Bridge

Product: 6800
organization: within



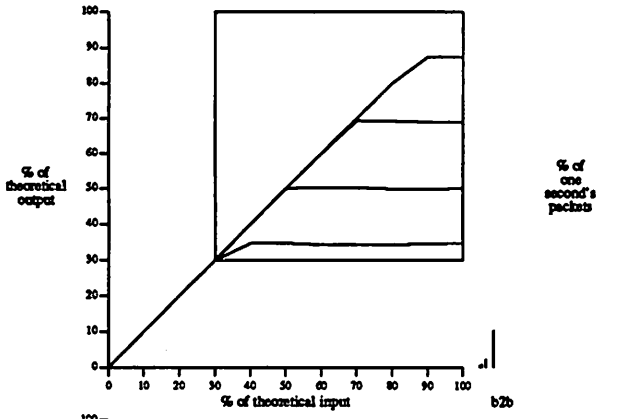

Vendor: NSC
Protocol: TCP/IP

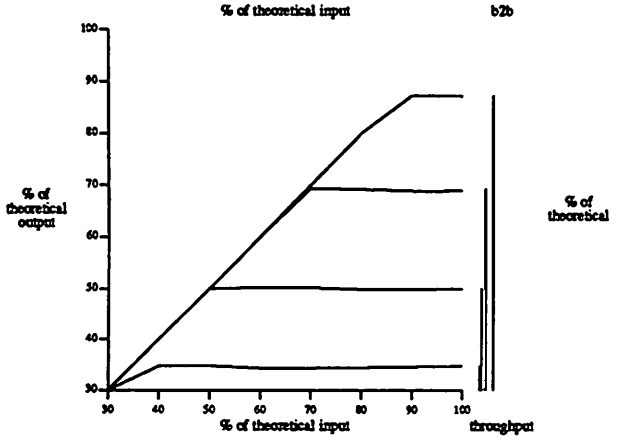
Product: 6800
organization: within



Vendor: Proteon
Protocol: AppleTalk II

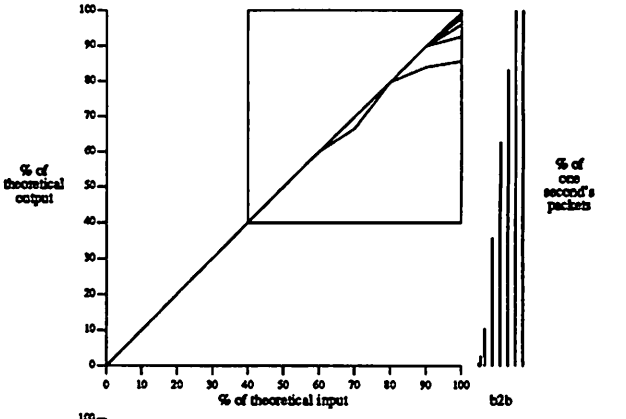
Product: CNXr_500
organization: between

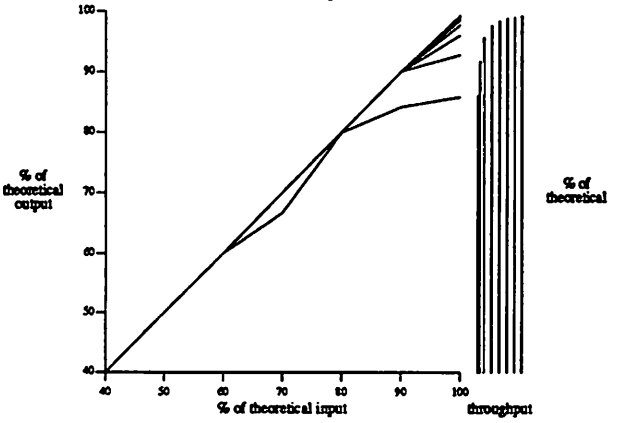



Vendor: Proteon
Protocol: TCP/IP

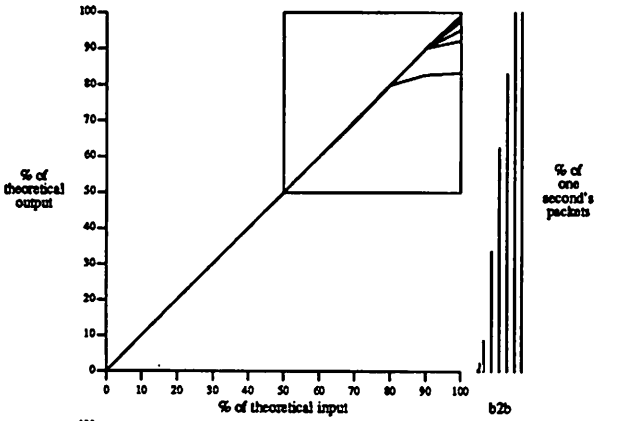
Product: CNXr_500
organization: between

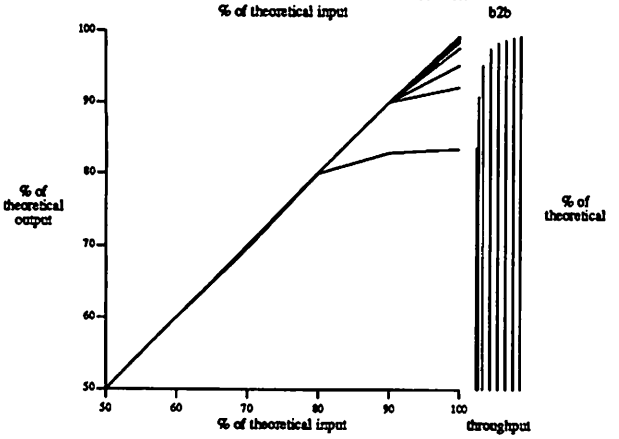



Vendor: Proteon
Protocol: Bridge

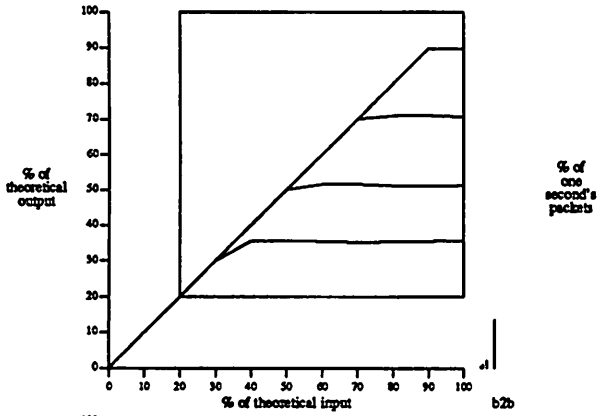
Product: CNXr_500
organization: between

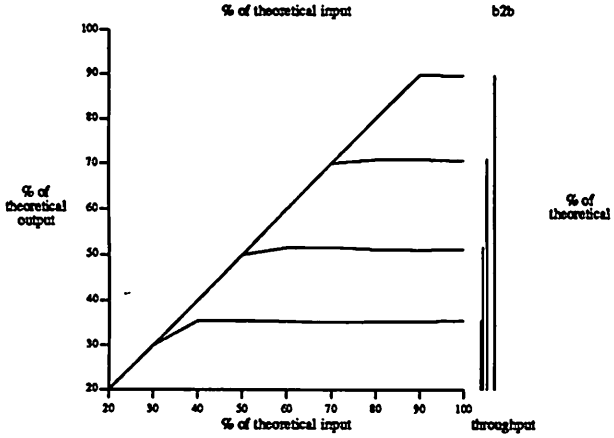


Vendor: Proteon
Protocol: AppleTalk II

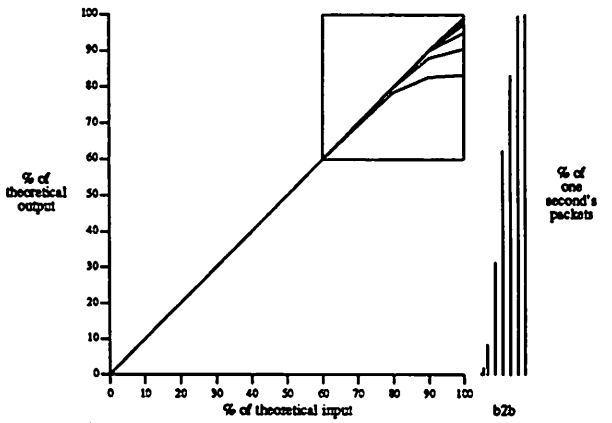
Product: CNX_500
organization: within

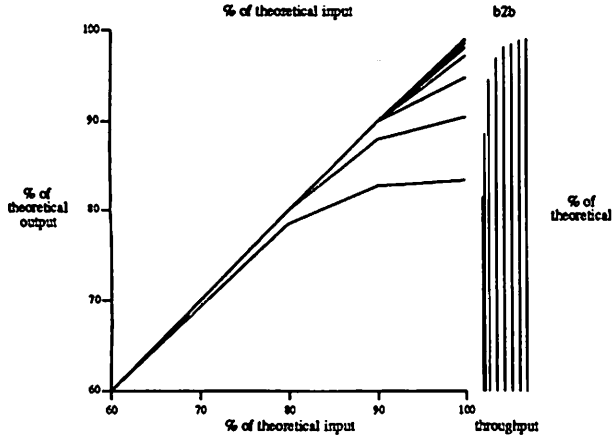



Vendor: Proteon
Protocol: Bridge

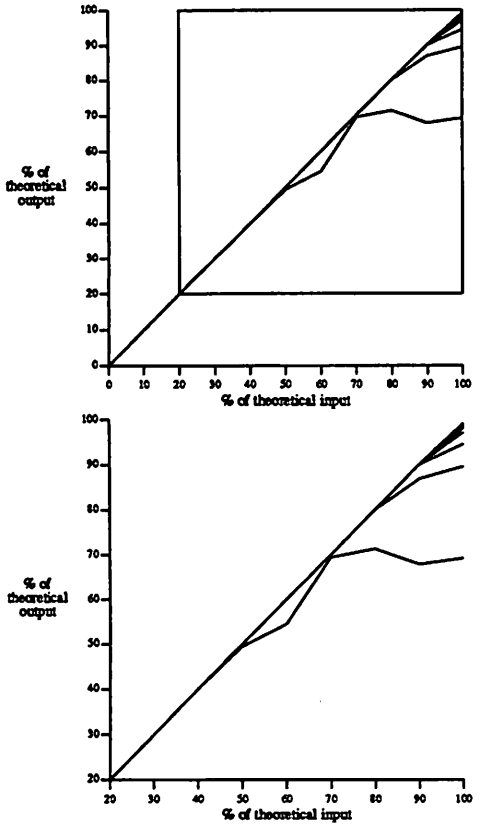
Product: CNX_500
organization: within



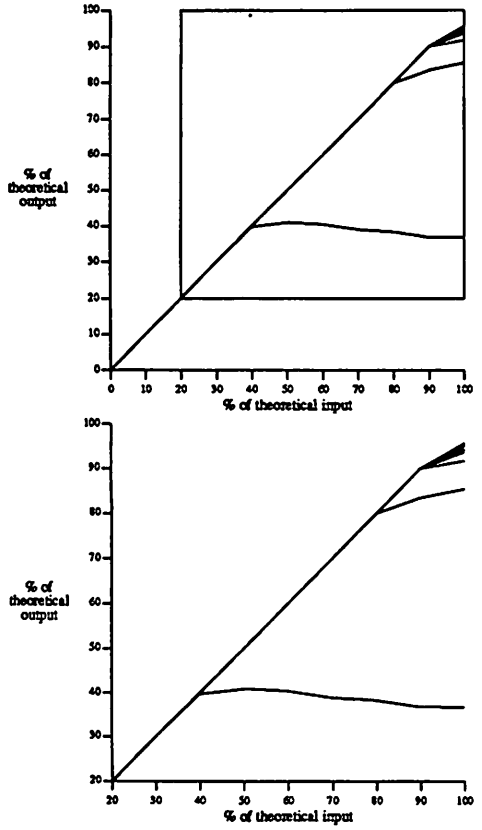
Vendor: Proteon
Protocol: TCP/IP

Product: CNXr_500
Streams: 2



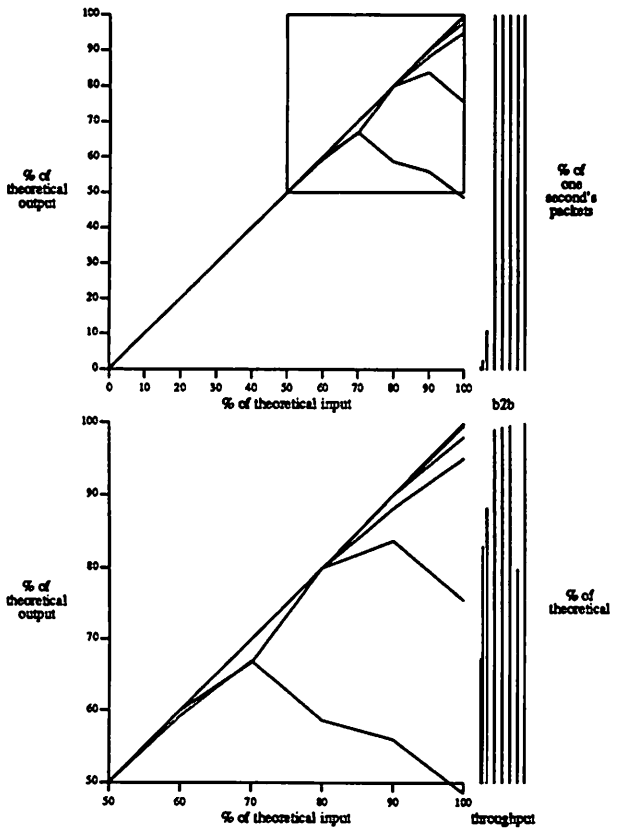

Vendor: Proteon
Protocol: TCP/IP

Product: CNXr_500
Streams: 3



Vendor: SUN_Microsystems
Protocol: TCP/IP

Product: SS2
organization: between



Vendor: Timeplex
Protocol: TCP/IP

Product: TIME/LAN
organization: between