NeTraMet

an Internet Accounting Meter

Nevil Brownlee
Computer Centre
The University of Auckland

IETF, Wed 30 Mar 94

Definitiions

- ***Accounting Meter***
  - Small host, attached to a network
  - Observes data packets
  - Builds a table of traffic flows
- ***Traffic Flows***
  - A *flow* is a stream of packets between two network hosts
  - Flows are bi-directional, *Source* and *Dest* are specified by the user
  - Each ‘end’ is described by a set of attributes
  - Flow has other attributes, e.g. octet and PDU counts, creation and last-use times

Addresses

- ***Address Attributes***
  - ***Adjacent* Address: link layer**
  - ***Peer* Address: network layer (IP, IPX, AppleTalk, DECnet, CLNS)**
  - ***Detail* Address: varies with protocol (IP port number)**
- ***Address Masks***
  - **Each address attribute also has a mask**
  - ***Mask* bits indicate which address bits are significant**

Rule Table (i)

- ***Rules*** **describe the flows of interest as a set of *patterns***
- **Meter matches these patterns against every packet it processes**
- **Each address attribute also has a *Mask***
- **Mask bits indicate which address bits are significant**
- **Packet processing –**
  - ***Classify*: match patterns to determine *flow.***
    No match –> ignore this packet
  - ***Search*: make sure this flow is in the meter's tables**
  - ***Count*: increment flow counters, set times**

Rule Table (ii)

- **A rule is a row in the meter's rule table**
- **Each rule has the form**
  { Attribute, Mask, Value, Action, Index }
- ***Attribute* is ANDed with *Mask* and compared to *Value*. Equal –> execute *Action*.**
  ***Index* is a parameter for Action**
- **Action can be –**
  - ***PushTo*: Index = next rule to test. (Remember matched rules)**
  - ***Succeed*: Stop matching, ignore the packet**
  - ***Count:* }**
  - ***Tally*: } Search, Insert, Update flow**
  - ***Aggregate*: }**

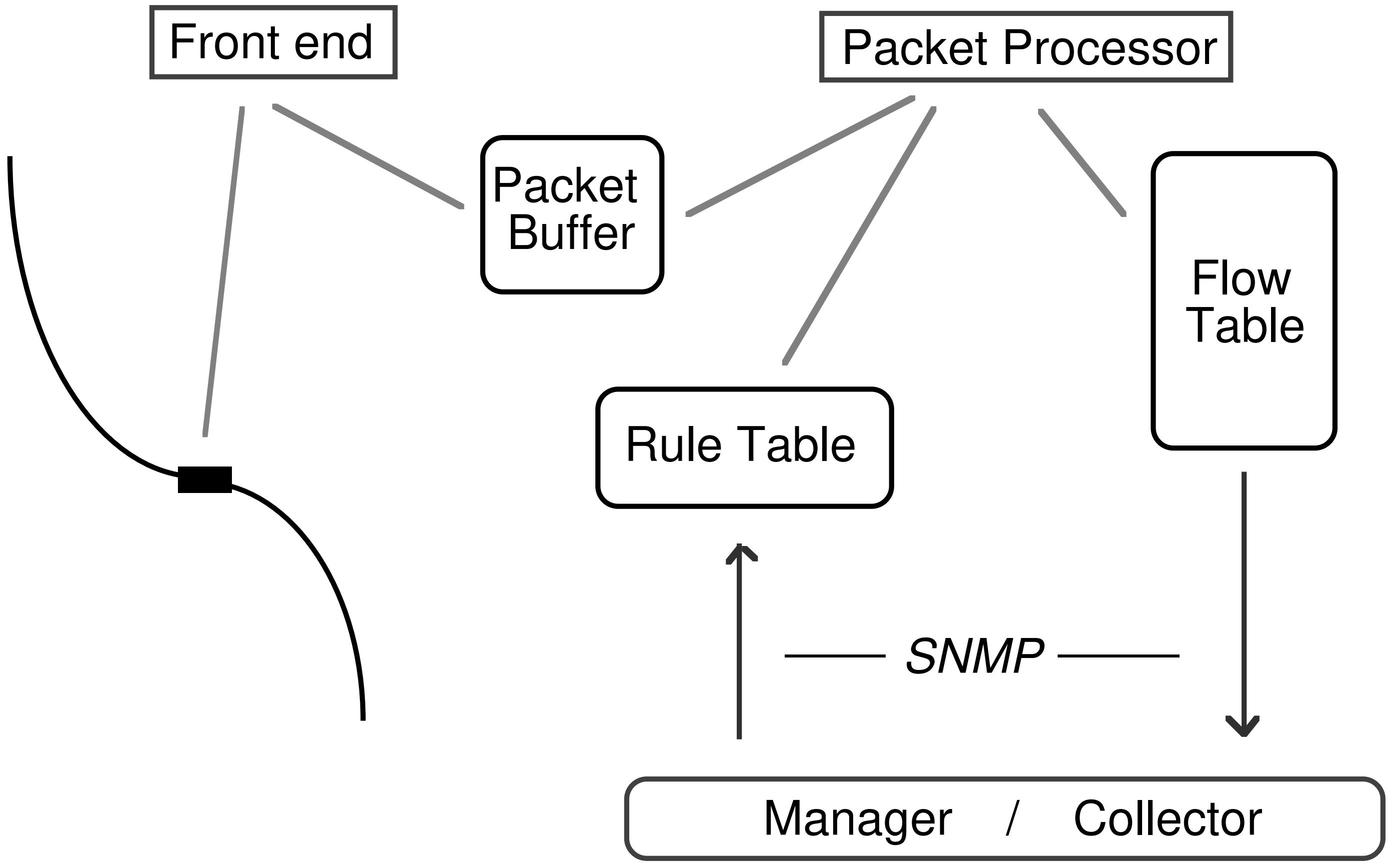
Count / Tally / Aggregate

- **Meter also has a table of Actions**
- **Each action is a prototype flow; masks and values may be specified for any attribute**
- ***Aggregate* –> Use masks and values from Action**
- ***Tally* –> Process packet as one of a set of Tally flows. Masks specified by Action, Values from the packet. e.g. test PeerAddress = Class B, tally with Class C mask**
- ***Count* –> Process this packet as one of a set of *Count* flows. Masks and values from matched rules**
- **NeTraMet builds a separate hash table for each Count and Tally action**

NeTraMet Extensions to Actions

- **Aim is to allow more powerful patterns, i.e. use less rules to describe more flows**
- ***Fail:*** **Causes match to fail. This allows mete to swap the packet's Source and Dest then retry the match**
- ***Assign:*** **}**
- ***Gosub:*** **} Allow rule-matching subroutines**
- ***Return n:*** **}**
- **Using subroutines reduced TuiaNet rule table from 850 rules and 220 actions to 225 rules and 10 actions**
- **Shorter rule tables provide faster pattern matching!**

NeTraMet Implementation

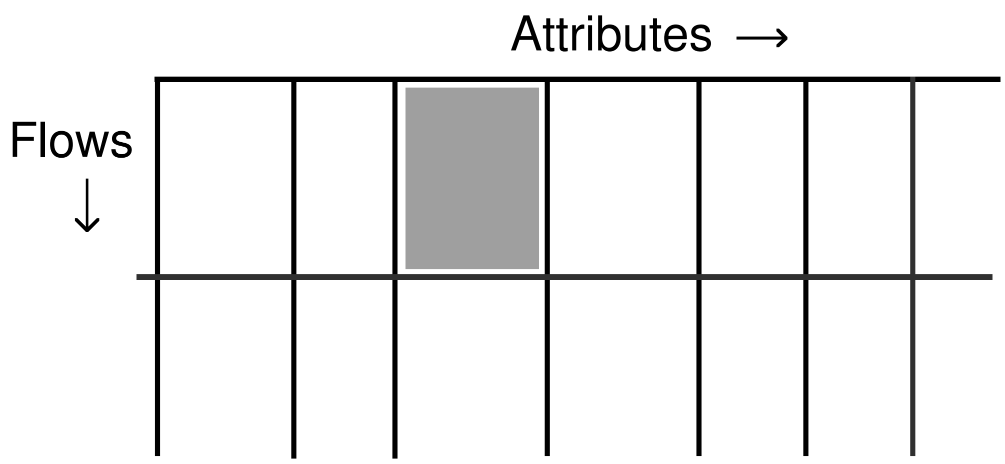


NeTraMet Features (i)

- **‘Creation’ and ‘Activity’ tables**
  - **Allow collection of flow data by several collectors at different intervals**
  - **SNMP object IDs for ‘creation’ and ‘activity’ table entries include an index and a time**
  - ***time* is in same form as the meter's UpTime, i.e. centiseconds from meter startup**
  - **Meter returns the first SNMP object with both time and index greater than the requested OID values**
  - **This allows a collector to find all the flows created or active after a specified time**

NeTraMet Features (ii)

- **Bulk Flow Data Collection**
  - **Flow Table is a 2-D array: Attributes x Flows**
  - **Meter uses an ASN.1 opaque object (‘column blob’) to collect part of an attribute column**



  - **Collector can read specified set of attributes for flows active after a given time – this minimises SNMP traffic**

NeTraMet Features (iii)

- **Dynamic Memory Management**
  - **Meter maintains table of collector IP addresses and collection times**
  - **Incremental garbage collector searches flow table for idle flows, i.e. flows which haven't been active since last collection**
  - **Space used by idle flows is recovered**
  - ***InactivityTimeout* variable specifies minimum lifetime for flows**
  - ***HighWaterMark* variable warns meter garbage collection should be speeded up. Meter does this by reducing InactivityTimeout**

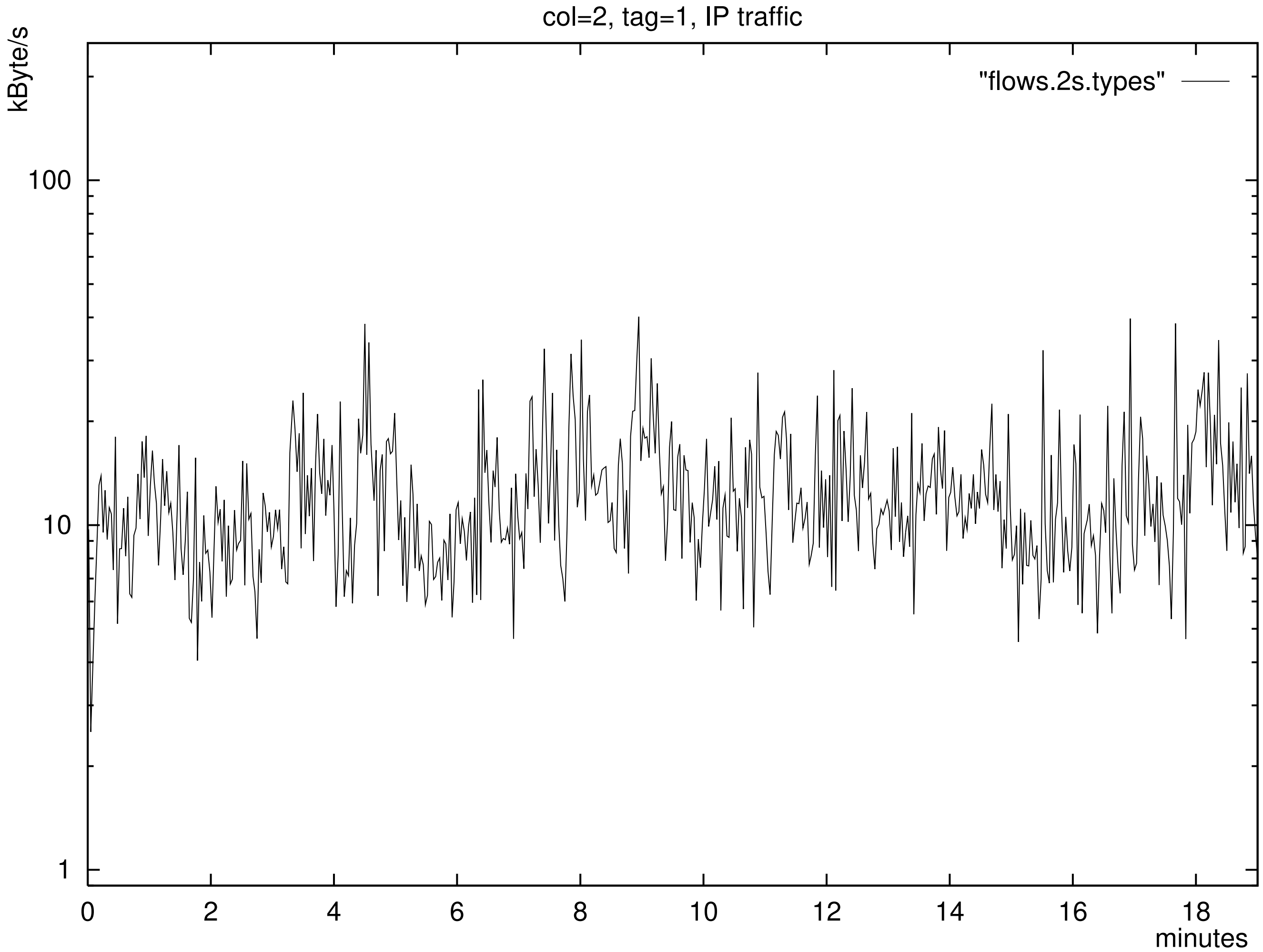
NeMaC: Manager and Collector

- **Reads (ASCII text) rule file, checks it for consistency, downloads it to meters**
- **Collects flow data from meter at specified intervals**
- **Can handle a list of meters, each with its own rule file**
- **Simple flow data file format – normalised matrix of attributes vs flows**
- **File lines starting with # hold management information, e.g. attribute names (Format), sample times (Time)**
- **No provision for starting new files at intervals, e.g. daily**

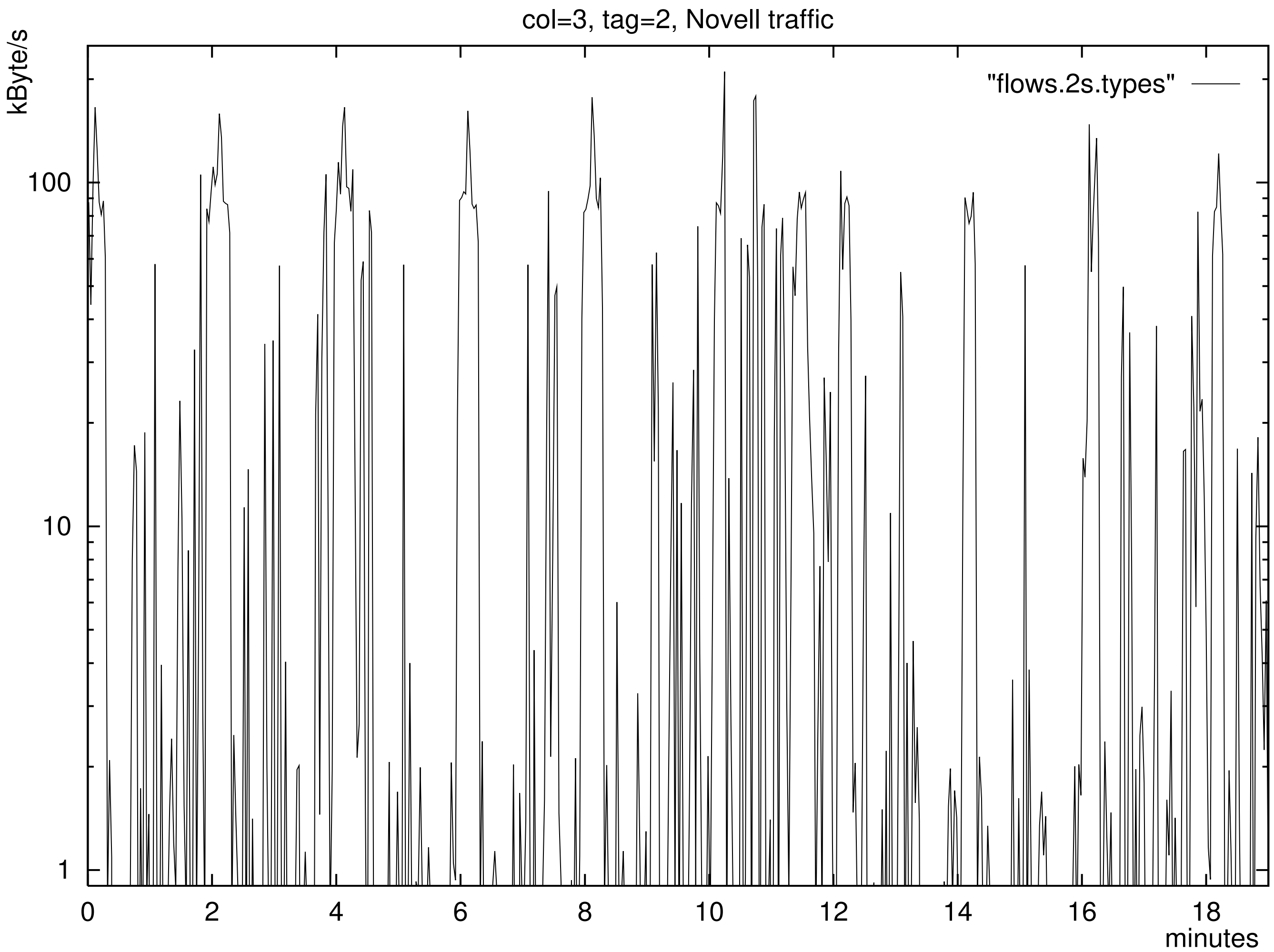
Usage Example (i): LAN Traffic, 2s samples

- Default rule set classifies traffic by Peer type, producing an aggregate flow for each type
- Flows were collected at 2-second intervals on a moderately busy LAN, with IP, IPX, EtherTalk, DECnet and VAX Cluster traffic
- Flow data file processed by *fd_filter* program to compute packet and byte rates, then by *fd_extract* to produce data file for plotting
- Plots produced using gnuplot

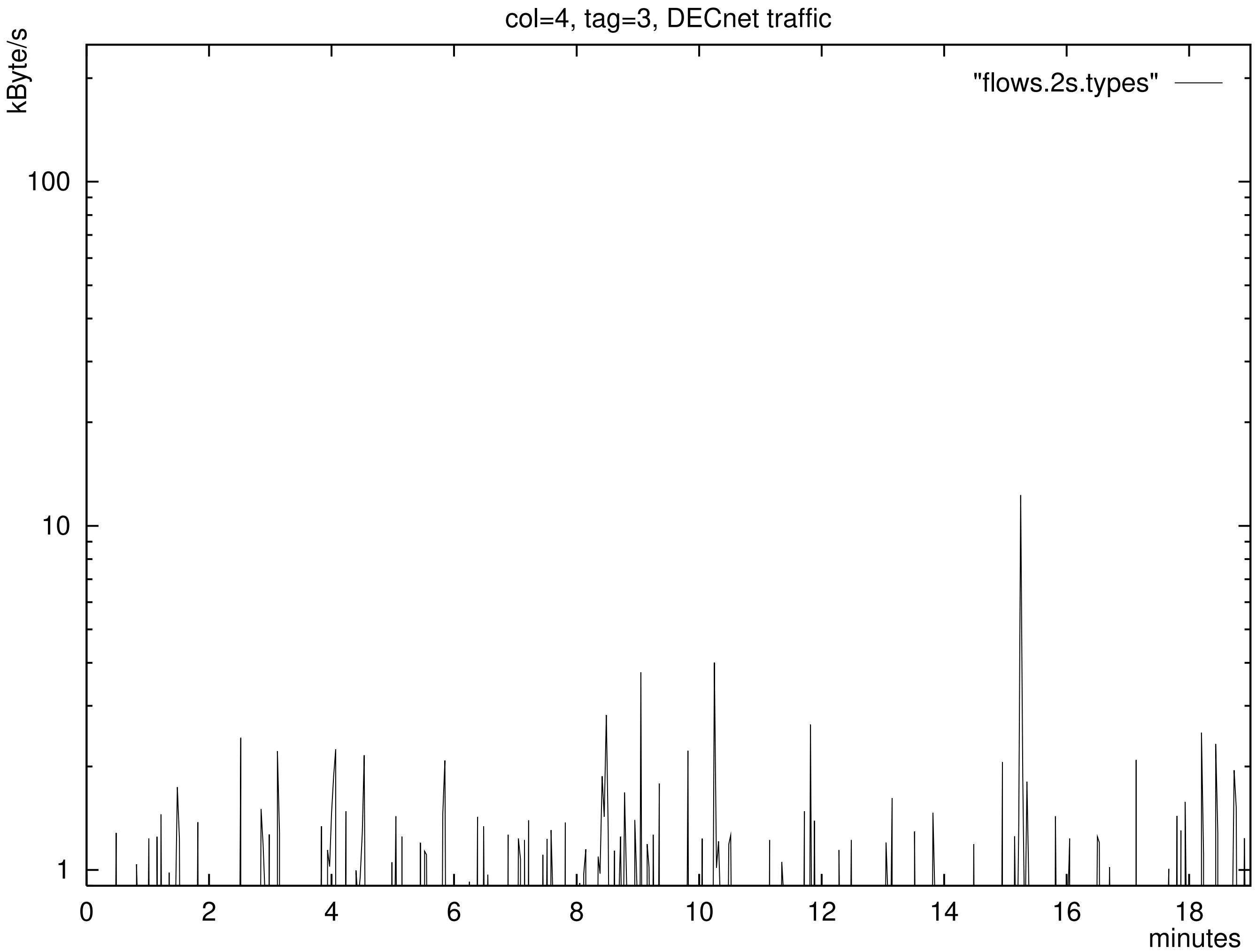
Plot of IP traffic on a LAN (2s samples)



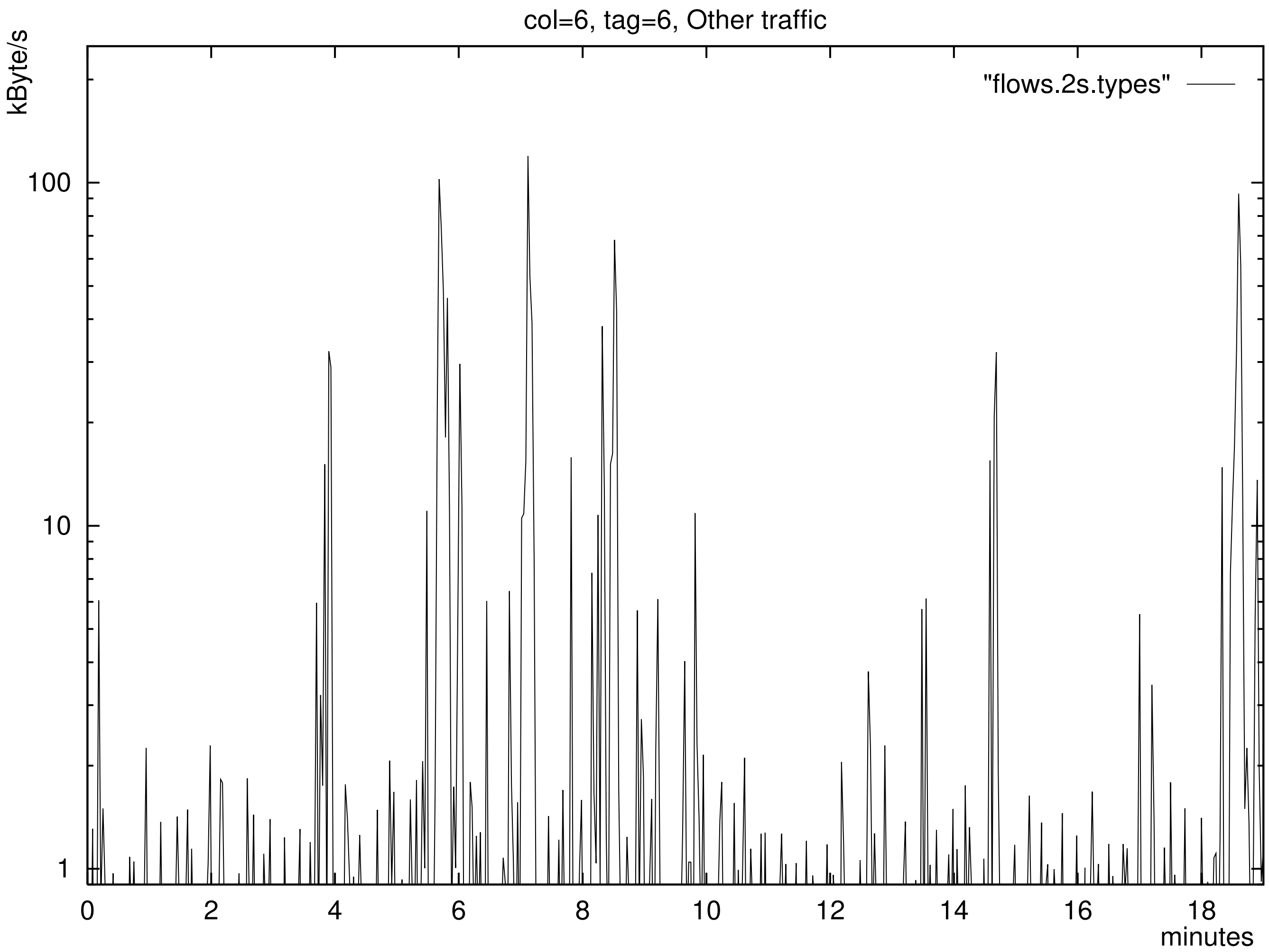
Plot of Novell traffic on a LAN (2s samples)



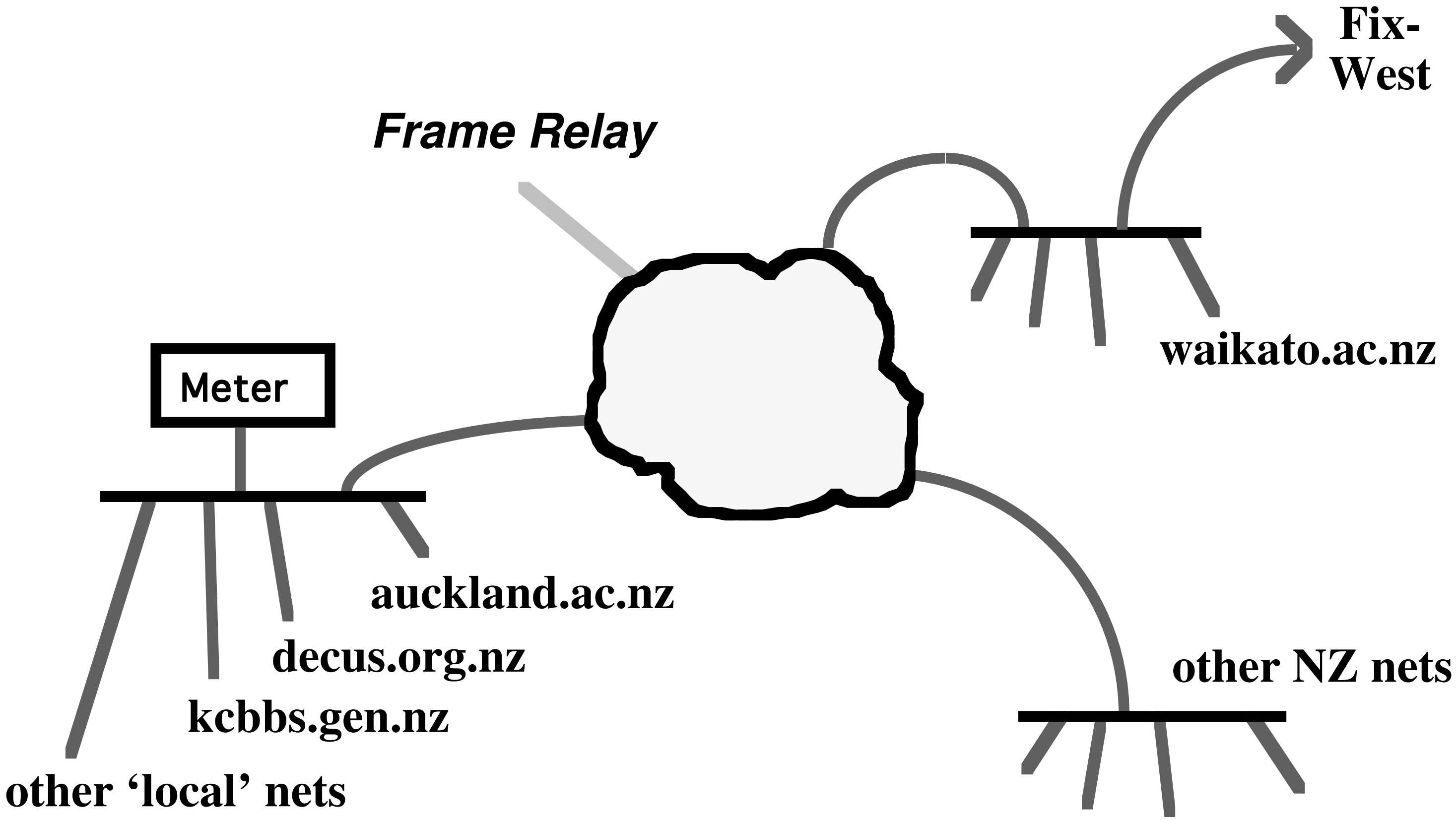
Plot of EtherTalk traffic on a LAN (2s samples)



Plot of Other traffic on a LAN (2s samples)

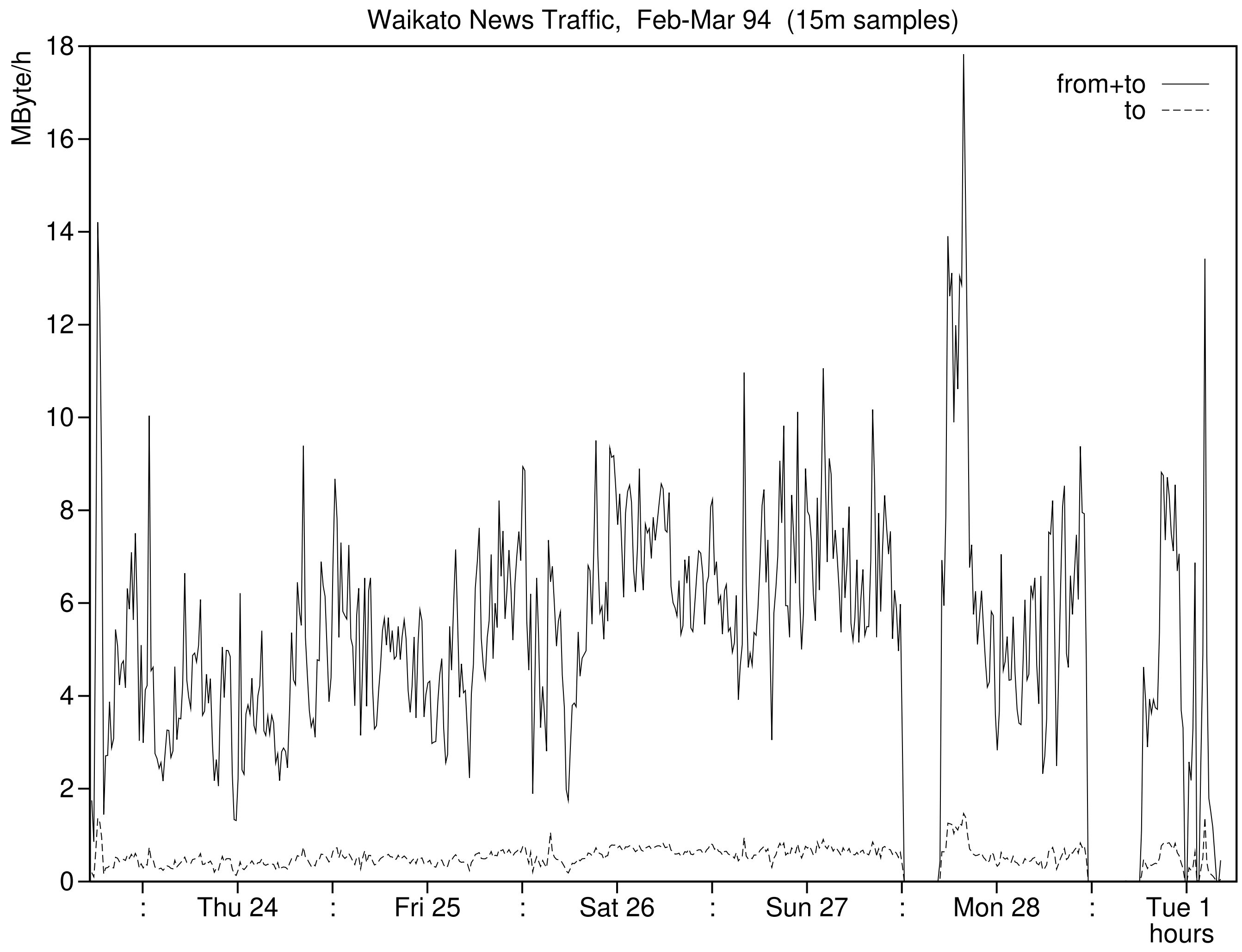


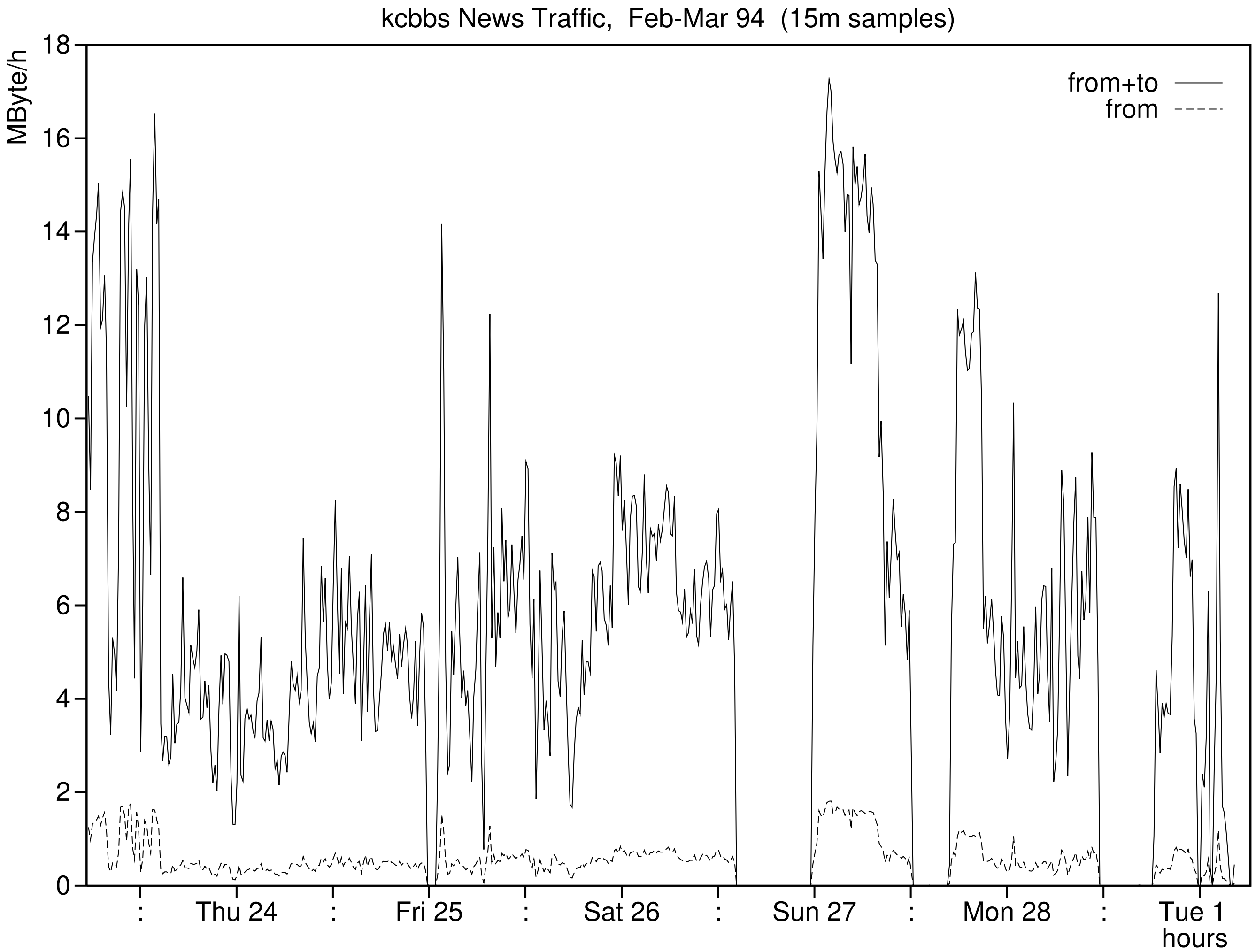
Usage Example (ii):
TuiaNet Traffic, 15m samples

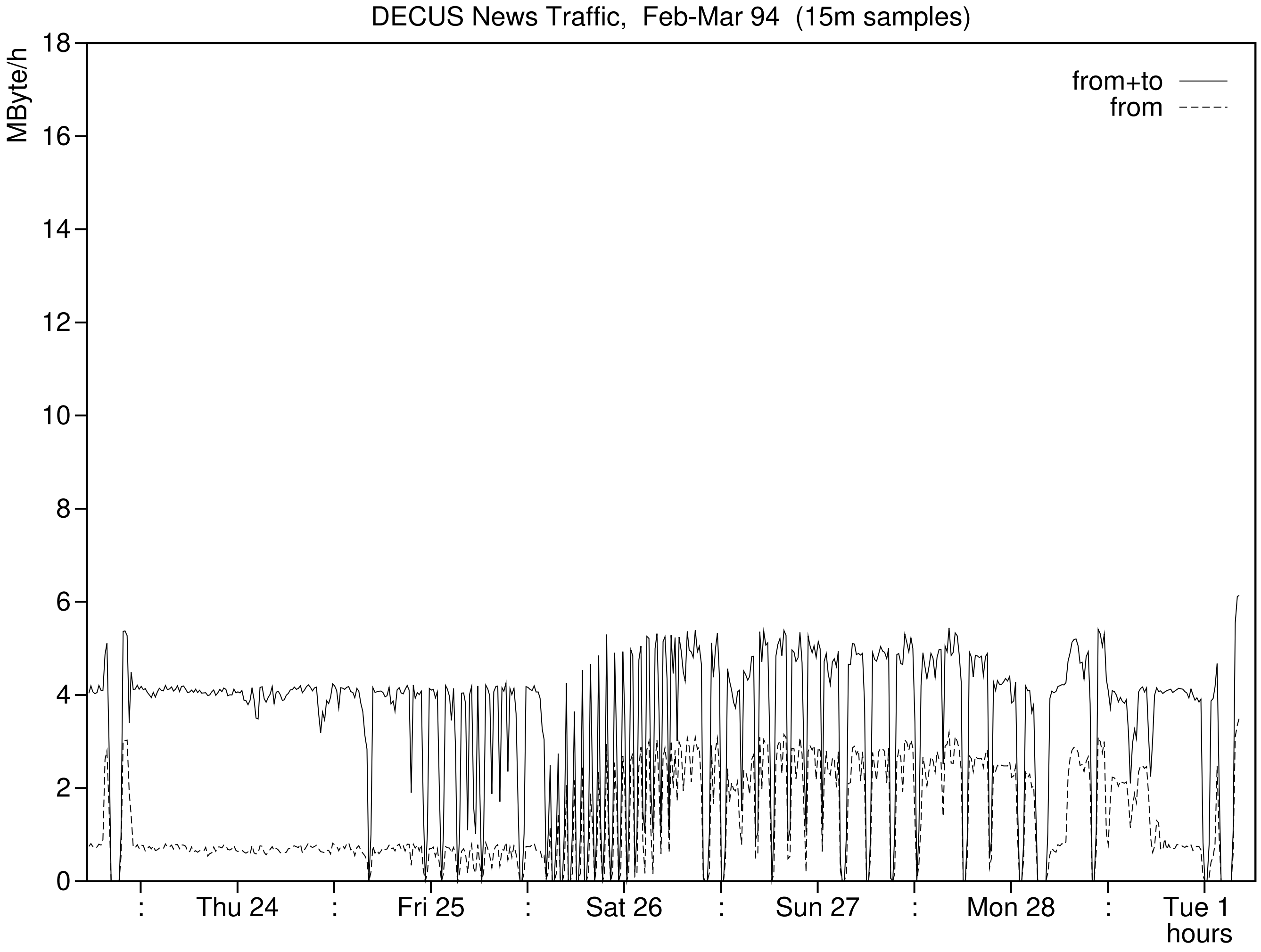


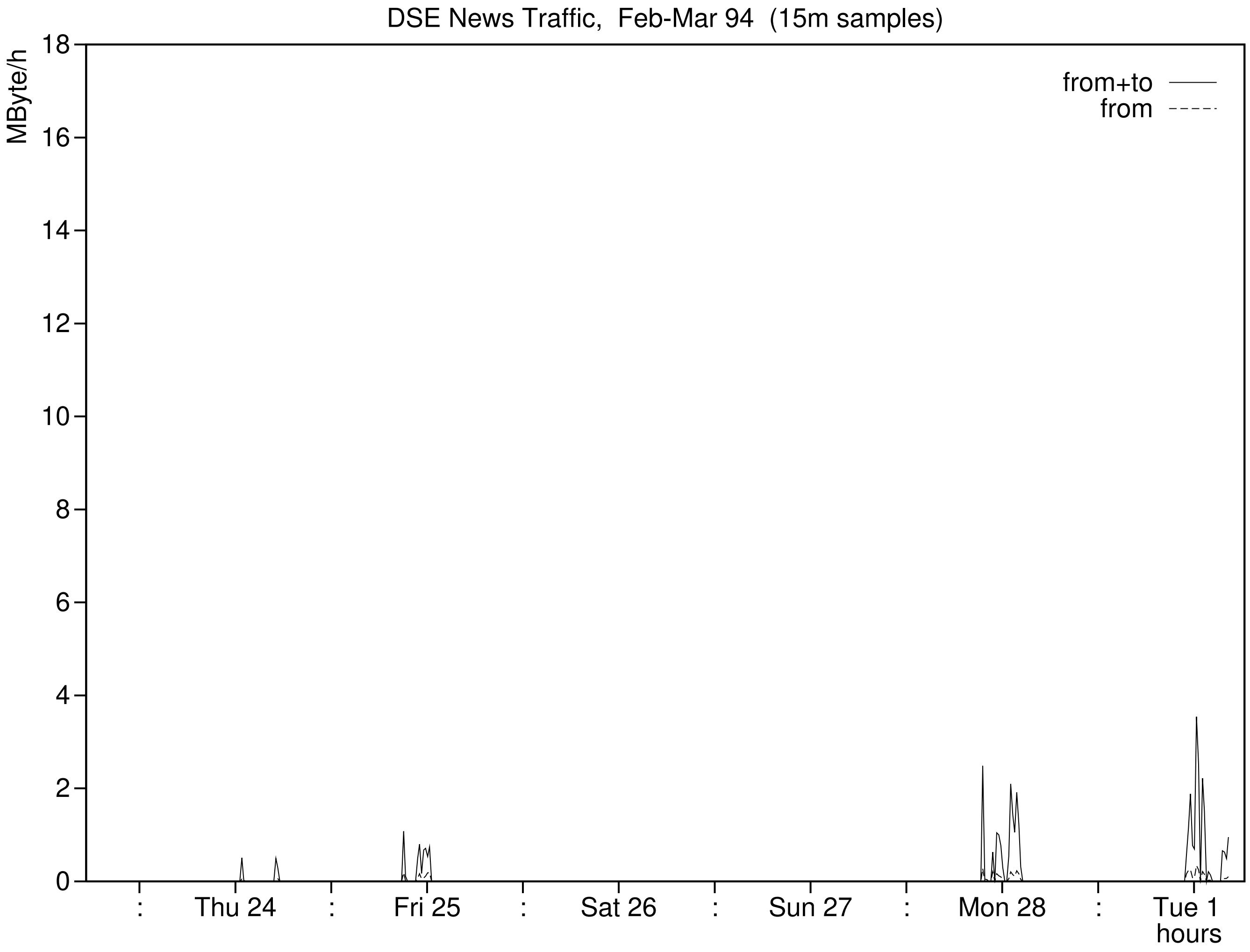
TuiaNet Rule Table

- **Classes of flows:**
  - **Auckland–Auckland** **ignore**
  - **Auckland–Local** **count Source–Dest**
  - **Auckland–NZ** **count Source–Tuia site**
  - **Auckland–World** **count Source–World**
  - **Local–NZ** **count Source–NZ**
  - **Local–World** **count Source–World**
- **Traffic types (by IP port number):**
  - **ICMP** **FTP Control**
  - **NNTP (News)** **Local FTP data**
  - **SMTP (Mail)** **Remote FTP data**
  - **DNS (Domain)** **Other TCP**
  - **Telnet** **Other UDP**
  - **Local–World** **count Source–World**


Waikato News Traffic, Feb-Mar 94 (15m samples)
MByte/h
from+to
to
0
2
4
6
8
10
12
14
16
18
Thu 24
Fri 25
Sat 26
Sun 27
Mon 28
Tue 1
hours


kcbbs News Traffic, Feb-Mar 94 (15m samples)
MByte/h
from+to
from
0
2
4
6
8
10
12
14
16
18
Thu 24
Fri 25
Sat 26
Sun 27
Mon 28
Tue 1
hours


DECUS News Traffic, Feb-Mar 94 (15m samples)
MByte/h
from+to
from
0
2
4
6
8
10
12
14
16
18
Thu 24
Fri 25
Sat 26
Sun 27
Mon 28
Tue 1
hours


DSE News Traffic, Feb-Mar 94 (15m samples)
MByte/h
18
16
14
12
10
8
6
4
2
0
from+to
from
Thu 24
Fri 25
Sat 26
Sun 27
Mon 28
Tue 1
hours

Implementation Status

- Reported performance
  - 10 MHz AT handles peaks of 1200 to 1600 pps
  - 33 MHz 486 handles peaks of about 3000 pps
- Systems

– DOS	NeTraMet	**
– Windows	–	**
– SunOS	NeTraMet	NeMaC
– Solaris	*	NeMaC
– Irix	**	NeMaC
– Ultrix	*	NeMaC
– HPUX	*	NeMaC

NeTraMet FTP Distribution Sites

Site	Directory
ftp.auckland.ac.nz	iawg/NeTraMetat
(any time between 0800 and 2000 GMT)	
ftp.near.net	pub/NeTraMet
ftp.delmarva.com	pub/nms
ftp.funet.fi	pub/networking/management/NeTraMet
ftp.EU.net	network/NeTraMet