Fast Local Internet Protocol

FLIP

Frans Kaashoek
Robbert van Renesse
Hans van Staveren

Leendert van Doorn
<leendert@cs.vu.nl>

vrije Universiteit *amsterdam*

OUTLINE

- **Protocol requirements**
- **Fast Local Internet Protocol**
- **Amoeba and FLIP**
- **Summary**

PROTOCOL ISSUES IN A DISTR. SYST.

- **Transparency**
- **Remote procedure call (RPC)**
- **Group communication**
- **Security**
- **Network management**

WHY A NEW PROTOCOL?

- **Stream protocols (TCP,TP4)**
  - **Latency involved in connection setup**
- **Datagram protocols (IP)**
  - **Identify hosts instead of processes**
  - **Network management**
  - **Transparency**
  - **Solutions: ICMP, ARP, RARP, IGMP, VMTP**
- **Better to start from scratch ...**

FLIP (Fast Local Internet Protocol)

- **FLIP is a connectionless (datagram) protocol**
- **FLIP is specifically designed for:**

  Transparency, RPC, Group communication, Security, Network management

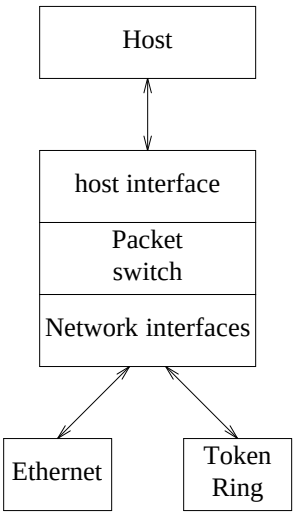
Level	OSI	TCP/IP	FLIP
7	Application	User-defined	User-defined
6	Presentation	Not used	AIL
5	Session	Not used	RPC and Group
4	Transport	TCP or UDP	Not needed
3	Network	IP	FLIP
2	Data Link	E.g., Ethernet	E.g., Ethernet
1	Physical	E.g., Coax cable	E.g., Coax cable

FLIP BASICS

- **Each FLIP entity has a 64-bit identifier (NSAP)**
  - **NSAPs are location independent**
  - **NSAP provides 256 56-bit address spaces**
  - **NSAP 0 is broadcast address**
- **Entities choose their own NSAP**
  - **NSAPs are chosen at random**
    (at-most-once, difficult to forge)

FLIP BOX

- **A machine is connected to internetwork by a FLIP box:**



- **Host interface is just another network**
- **The packet switch maintains a routing table**
- **Each machine is a gateway**

HOST INTERFACE

- **Interface betweeen host and packet switch:**

  flip_init(id, rcv, notdlvr) → ifno
  flip_end(ifno)
  flip_register(ifno, priv-addr) → ep
  flip_unregister(ifno, ep)
  flip_unicast(ifno, msg, flgs, dst, ep, len)
  flip_multicast(ifno, msg, flgs, dst, ep, len, ndst)
  flip_broadcast(ifno, msg, flgs, ep, hops)

  receive(id, frag descr)
  notdelivered(id, frag descr)

- **FLIP addresses:**
  - **pub-addr = F(low 56-bits priv-addr)**
  - **F(address) = 0 iff address = 0**

FLIP PROTOCOL

- **FLIP header:**

7	6	5	4	3	2	1	0
Max HopCnt		Actual HopCnt		Res.	Flags	Type	Vers.
0	Destination Address						
0	Source Address						
Length				Message Identifier			
Total Length				Offset			

- **FLIP header types:**

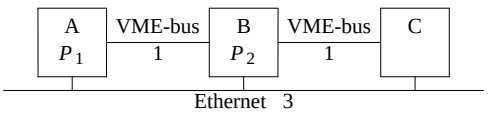
Type	Function
LOCATE	Find network location of NSAP
HEREIS	Reply on LOCATE
UNIDATA	Send a fragment point-to-point
MULTIDATA	Multicast a fragment
NOTHERE	Destination NSAP is unknown
UNTRUSTED	Unreachable over trusted networks

ROUTING

- **Every FLIP box is a router**
- **FLIP finds its routes dynamically**
- **Routing is based on:**
  - **Network weight**
  - **Trusted networks**
- **Each FLIP box maintains routing tuples**

  (address, network, location, hops, trusted, age, local)
- **Routing information is only a hint**

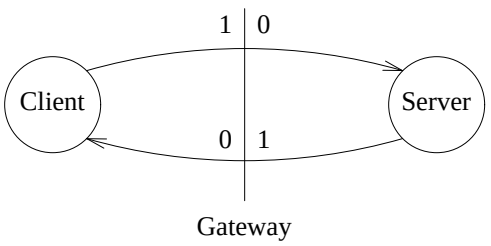
ROUTING EXAMPLE



- P_1 **does RPC with** P_2**:**
  - A's RPC layer calls flip_unicast(P_2)
  - FLIP does LOCATE P_2 with maxhops = 1
  - B receives LOCATE for P_2
  - B saves (P_1, hops, and A's VME-bus address)
  - B replies with HEREIS
- P_2 **migrates to C**
  - B's RPC layer calls flip_unregister(*private* (P_2))
  - A sends UNIDATAs to B
  - B replies with NOTHERE
  - A calls notdeliver(message)
  - Restart locate of P_2

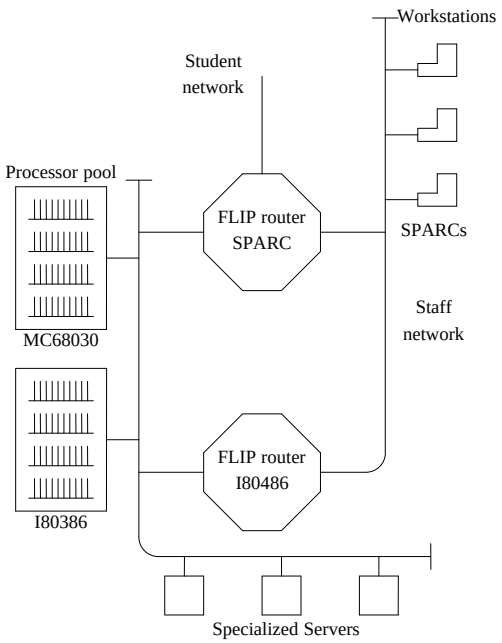
SECURITY

- **Trusted networks (TN)**
  - **Only route over TNs**
  - **Flag for noting TN boundary crossings**

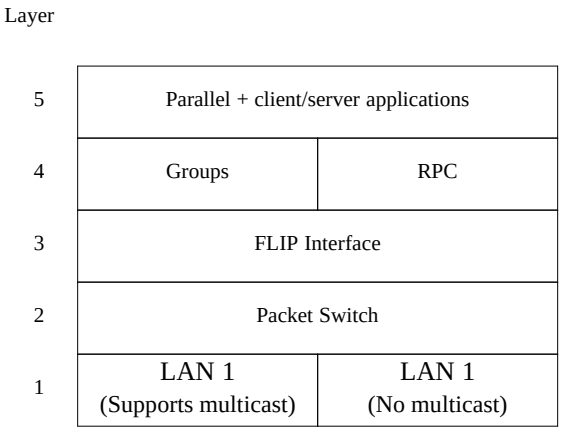


- **Confine certain messages to TN**
  - **On TN no need for encryption**
  - **Efficient security systems**
- **Requires administration**
- **But, what is trust ...**

FLIP UNDER AMOEBA



INTEGRATION WITH AMOEBA



- **Supports both RPC and group communication**
- **Supports LANs with/without hardware multicast**
- **Programs are portable to both kinds of LANs**
- **Useful for building small scale WANs**

TRANSPARENCY

- **Holy grail of Amoeba**
- **Two level naming scheme**
  - **Ports (capabilities)**
  - **FLIP addresses**
  - **Mapping between ports and FLIP addresses**
- **Support for large messages** $2^{32}-1$

AMOEBA COMMUNICATION

- **At-most-once RPC**
  - **FLIP address 0 for broadcasting**
  - **New FLIP address for every server**
- **Group communication**
  - **FLIP address 0 for broadcasting**
  - **FLIP address for each member**
  - **FLIP address for group**
  - **Uses** flip_multidata **interface**

    (FLIP uses multicast when available)

SECURITY

- **Security**
  - **Experimental secure RPC using TN**
  - **Traditional authentication system**
- **Network management**
  - **Network reconfigurations**
- **Wide-area communication**
  - **Encapsulated FLIP over TCP/IP**
  - **Small scale wide-area distributed system**

    (VU, CWI, UVA), (VU, Cornell)

IMPLEMENTATION AND PERFORMANCE

- **Runs on SPARC station, 68000 family, and 80386**
- **Network interfaces for**

  Ethernet, shared memory, and TCP/IP
- **RPC (Sun3/60s connected by Ethernet):**
  - **Delay for null RPC 2.1 msec**
  - **Bandwidth 1061 Kbytes/sec**
- **Group communication (Force30s connected by Ethernet):**
  - **A null message to 30 processes 2.7 msec**
  - **Per group 800 null messages per sec**

SUMMARY OF FLIP

Requirements	**FLIP support**
Transparency	• FLIP addresses are location independent • A FLIP-box performs routing • Messages can be as large as $2^{32}-1$ bytes.
Efficient at-most-once RPC	• FLIP is an unreliable datagram protocol • FLIP can use a blast protocol for large messages • A process uses a new FLIP address after crash
Group Communication	• A FLIP address per group • Routing tables change dynamically • FLIP uses physical multicast, if possible • FLIP also works if multicast is not available
Security	• Source addresses cannot be forged • FLIP takes advantage of trusted networks
Network Management	• Every machine is a router • Routing tables are dynamically updated