Packet Header Analysis on the IXP 1200 Network Processor

Institute of Computer Science,
Foundation of Research
& Technology, Hellas
(ICS- FORTH)

I.Charitakis
D.Pnevmatikatos
E. Markatos

Distributed Systems Lab.,
CIS Dep. Univ of Pennsylvania,
Phila, USA

K.Anagnostakis

Outline

1. **Network Intrusion Detection Systems**
2. Network Processors
3. Packet Header Analysis on IXP 1200
4. Future Directions

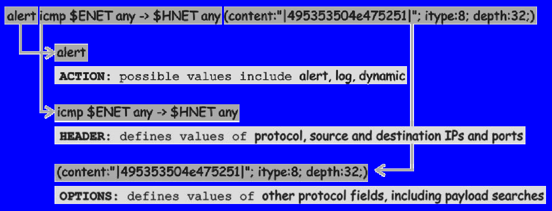
Network Intrusion Detection Systems (NIDS)

- Per packet processing
- Signatures describe dangerous packets
- Snort (www.snort.org)
  - Open Source
  - Widely deployed
  - State of the art

Snort Signatures

```
# (C) Copyright 2001,2002, Martin Roesch, Brian Caswell, et al.
#    All rights reserved.
# $Id: icmp.rules,v 1.18 2002/08/18 20:28:43 cazz Exp $
# ICMP RULES
#-----------
# Description:
# These rules are potentially bad ICMP traffic.  They include most of the
# ICMP scanning tools and other "BAD" ICMP traffic (Such as redirect host)
#
# Other ICMP rules are included in icmp-info.rules
alert icmp $ENET any -> $HNET any (itype:5;icode:1; )
alert icmp $ENET any -> $HNET any (itype:5;icode:0; )
alert icmp $ENET any -> $HNET any (itype: 4; icode: 0; )
alert icmp $ENET any -> $HNET any (itype: 8; icmp_id: 0; icmp_seq: 0; dsize:4; )
alert icmp $ENET any -> $HNET any (content:"|495353504e475251|"; itype:8;
                        depth:32;)
```

Signature Structure



alert icmp $ENET any -> $HNET any (content:"|495353504e475251|"; itype:8; depth:32;)

alert

ACTION: possible values include alert, log, dynamic

icmp $ENET any -> $HNET any

HEADER: defines values of protocol, source and destination IPs and ports

(content:"|495353504e475251|"; itype:8; depth:32;)

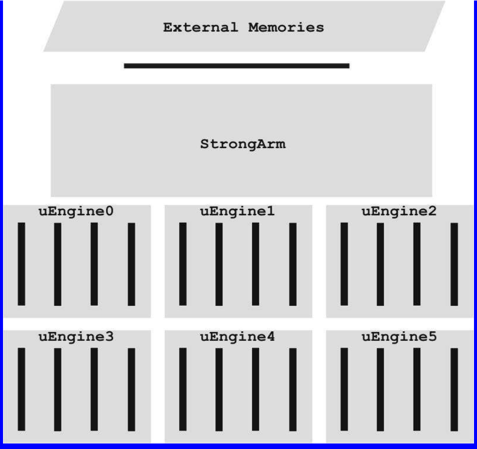
OPTIONS: defines values of other protocol fields, including payload searches

Outline

1. Network Intrusion Detection Systems
2. **Network Processors**
3. Packet Header Analysis on IXP 1200
4. Future Directions

Network Processors

- Flexibility and Performance
  - Speed : Application Specific Integrated Circuit
  - Flexibility : General Purpose Host Processor
- Extra hardware units
  - Exploit packet parallelism
  - Pipelining
- Difficult to program



Micro Engines

- A.k.a. uEngines
- Four h/w supported threads
  - Support of references
- 2 K instruction memory
- 128 general purpose registers

Network Processors

- Already studied for routing like applications
  - Modularity + Performance
    - Intel ACEs
    - VERA by Princeton University
    - Netbind by Columbia University
  - Modularity supports much easier programming
  - Comes at some cost on Performance
  - NIDS need Performance

Outline

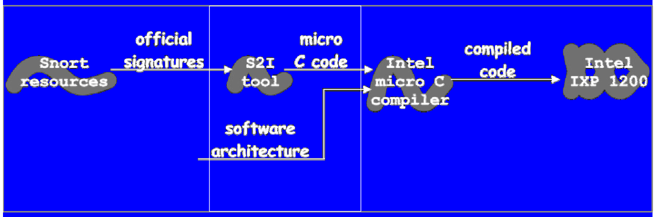
1. Network Intrusion Detection Systems
2. Network Processors
3. **Packet Header Analysis on IXP 1200**
   1. **Work Overview**
   2. Software Architecture
   3. The S2I tool
   4. Experiments
4. Future Directions

Work Overview : Objective

Facilitate deployment of IXP in NIDS

- Use the IXP for Packet Header Analysis
  - Investigate and propose programming methodology
  - Common tasks should be easy to perform
    - Updates on the active signatures

Use case example



Outline

1. Network Intrusion Detection Systems
2. Network Processors
3. **Packet Header Analysis on IXP 1200**
   1. Work Overview
   2. **Software Architecture**
   3. The S2I tool
   4. Experiments
4. Future Directions

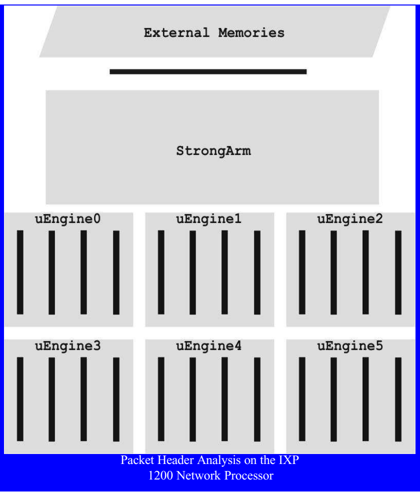
Software Architecture

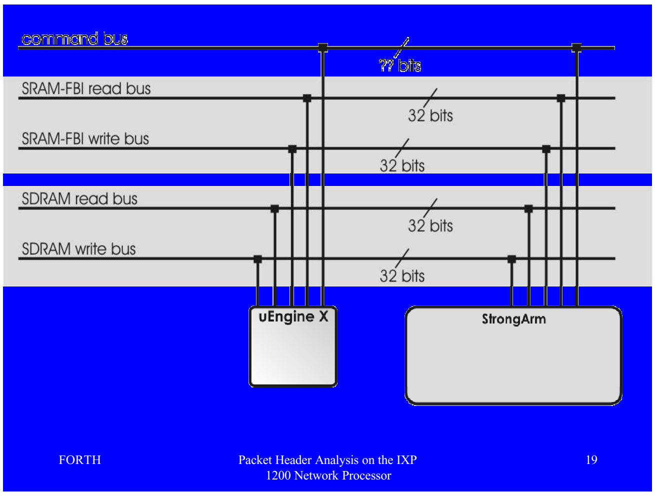
- Minimal infrastructure
  - Static Section – Never changes
  - IXP 1200 specific
  - Packet distribution to workers
- Primary Concerns
  - Space: minimal overhead in instructions and registers
  - Time: maximize *headroom* of each worker
    - » skip

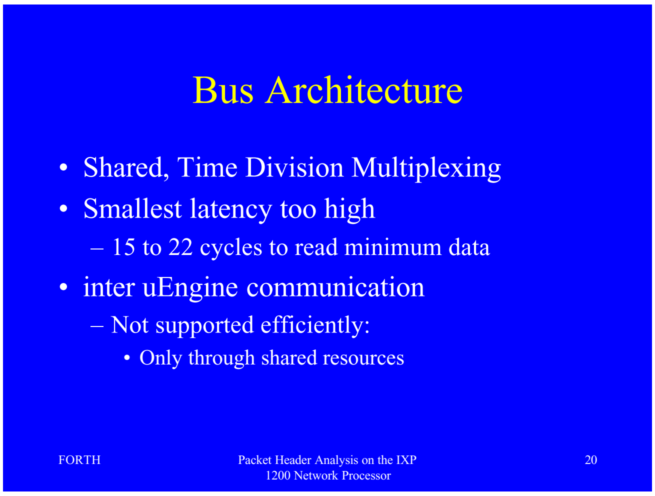
Basic design guidelines

- Independent packet processing
  - Minimize accesses to shared resources
  - Why ?

IXP 1200 Top View

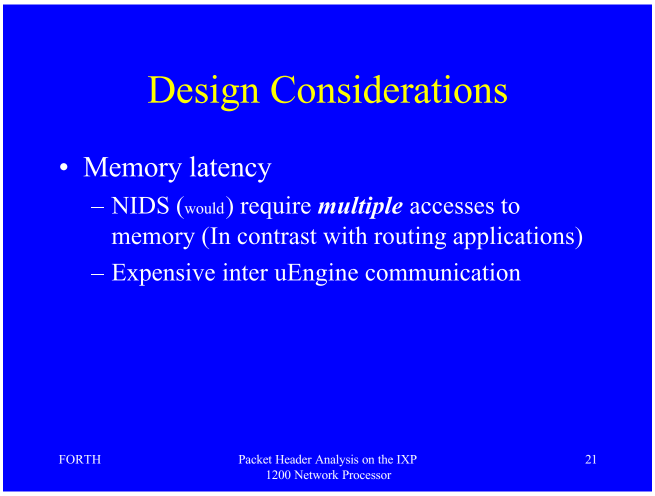






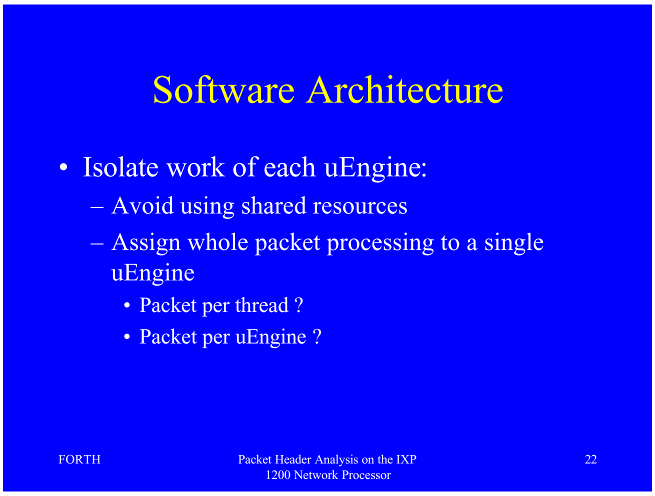
Bus Architecture

- Shared, Time Division Multiplexing
- Smallest latency too high
  - 15 to 22 cycles to read minimum data
- inter uEngine communication
  - Not supported efficiently:
    - Only through shared resources



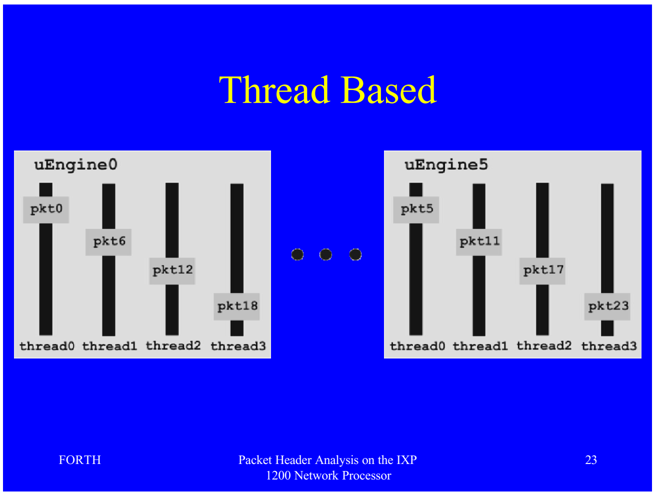
Design Considerations

- Memory latency
  - NIDS (would) require ***multiple*** accesses to memory (In contrast with routing applications)
  - Expensive inter uEngine communication

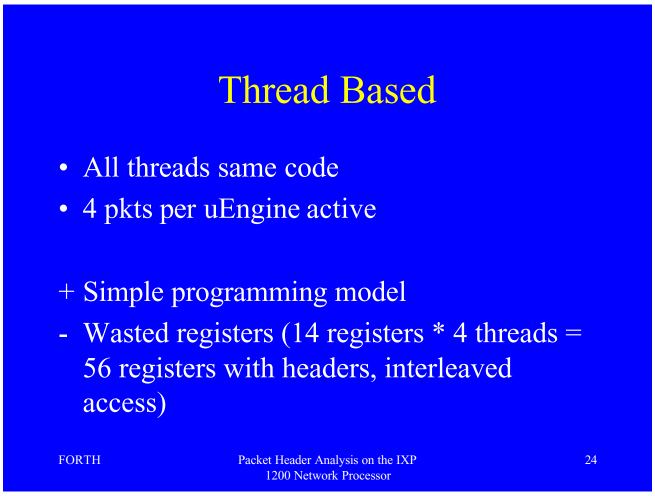


Software Architecture

- Isolate work of each uEngine:
  - Avoid using shared resources
  - Assign whole packet processing to a single uEngine
    - Packet per thread ?
    - Packet per uEngine ?



Thread Based

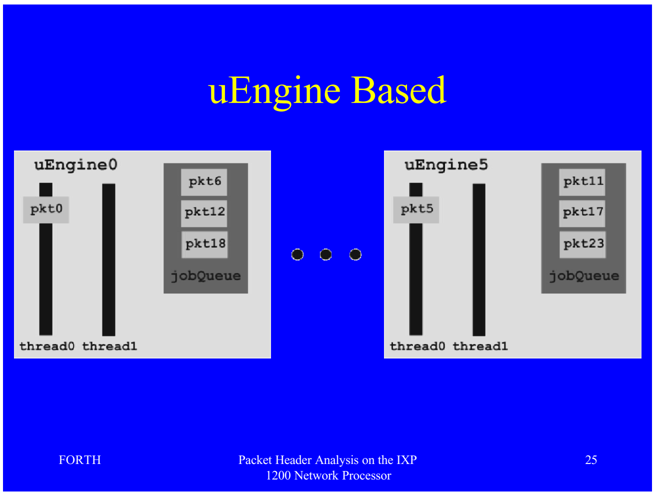


Thread Based

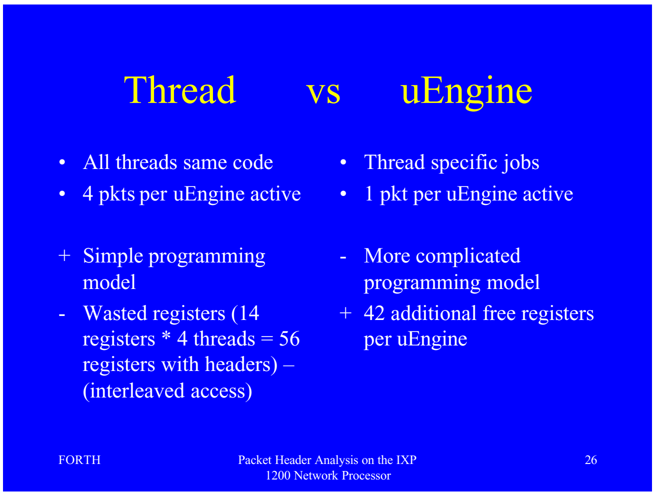
- All threads same code
- 4 pkts per uEngine active

+ Simple programming model

- Wasted registers (14 registers * 4 threads = 56 registers with headers, interleaved access)

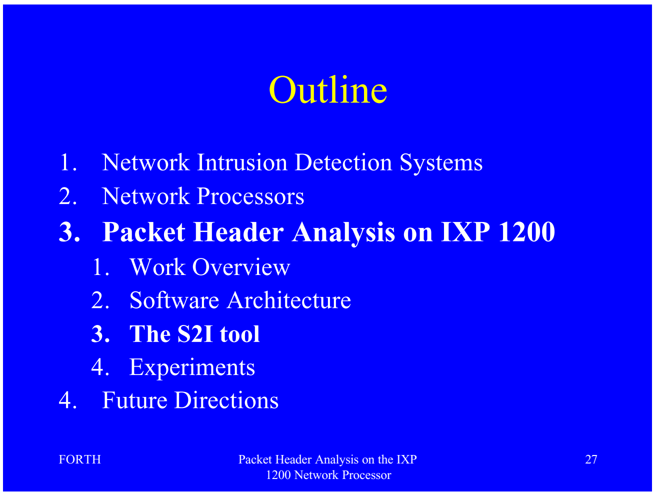


uEngine Based



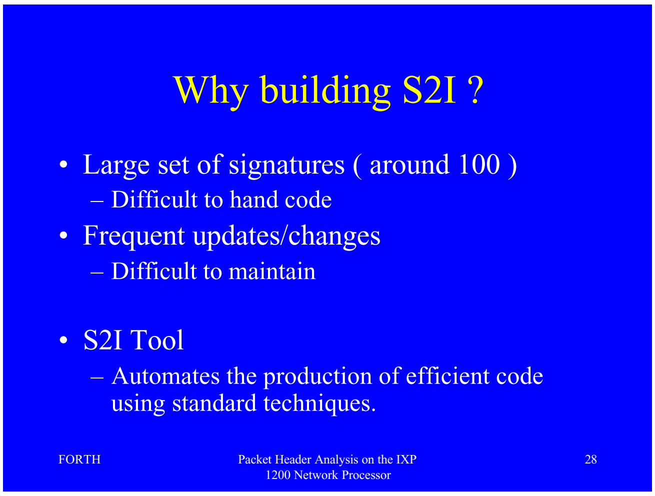
Thread vs uEngine

Thread	uEngine
• All threads same code	• Thread specific jobs
• 4 pkts per uEngine active	• 1 pkt per uEngine active
+ Simple programming model	- More complicated programming model
- Wasted registers (14 registers * 4 threads = 56 registers with headers) – (interleaved access)	+ 42 additional free registers per uEngine



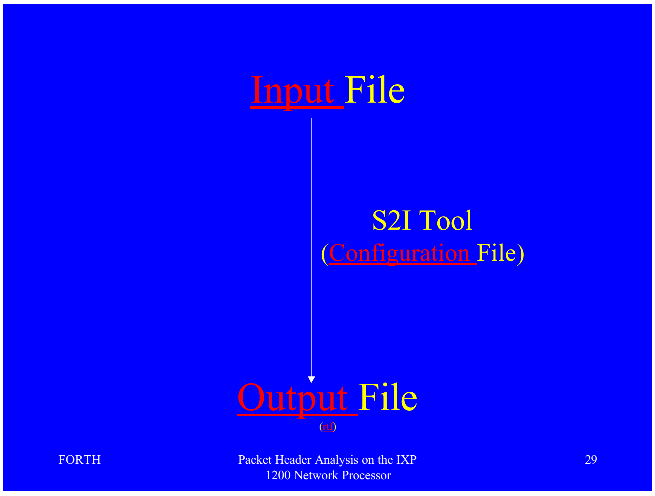
Outline

1. Network Intrusion Detection Systems
2. Network Processors
3. **Packet Header Analysis on IXP 1200**
   1. Work Overview
   2. Software Architecture
   3. **The S2I tool**
   4. Experiments
4. Future Directions



Why building S2I ?

- Large set of signatures (around 100)
  - Difficult to hand code
- Frequent updates/changes
  - Difficult to maintain
- S2I Tool
  - Automates the production of efficient code using standard techniques.



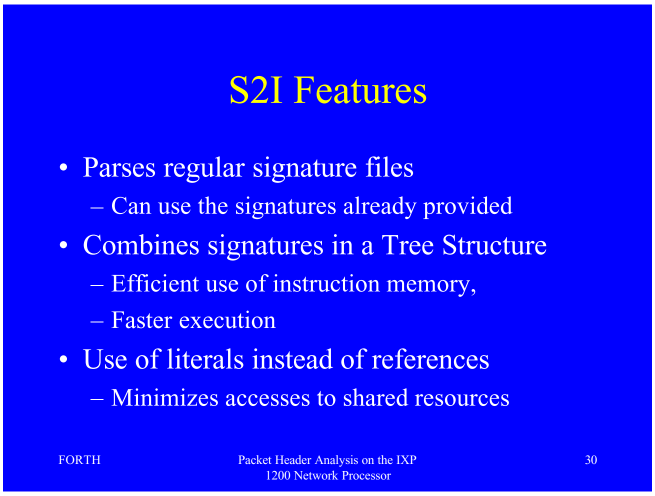
Input File

S2I Tool

(Configuration File)

Output File

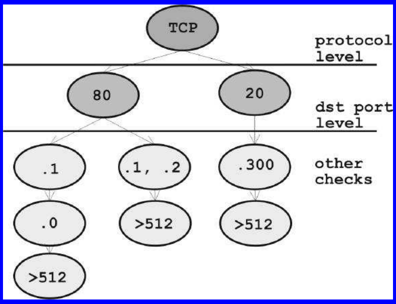
(rtf)



S2I Features

- Parses regular signature files
  - Can use the signatures already provided
- Combines signatures in a Tree Structure
  - Efficient use of instruction memory,
  - Faster execution
- Use of literals instead of references
  - Minimizes accesses to shared resources

Tree Representation



Use of Literals

- `If (protocol == **signature[i].protocol**){`

  `…`

  `}`

- `If (protocol == **6** ) {`

  `…`

  `}`

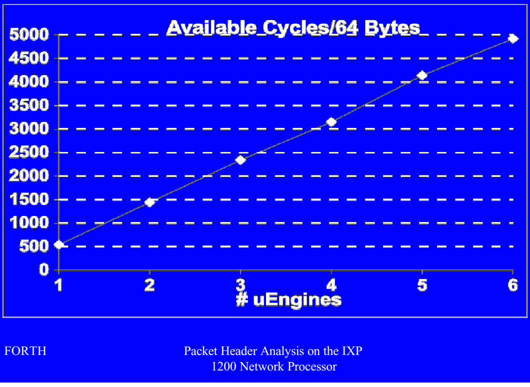
Outline

1. Network Intrusion Detection Systems
2. Network Processors
3. **Packet Header Analysis on IXP 1200**
   1. Work Overview
   2. Software Architecture
   3. The S2I tool
   4. **Experiments**
4. Future Directions

Experiments

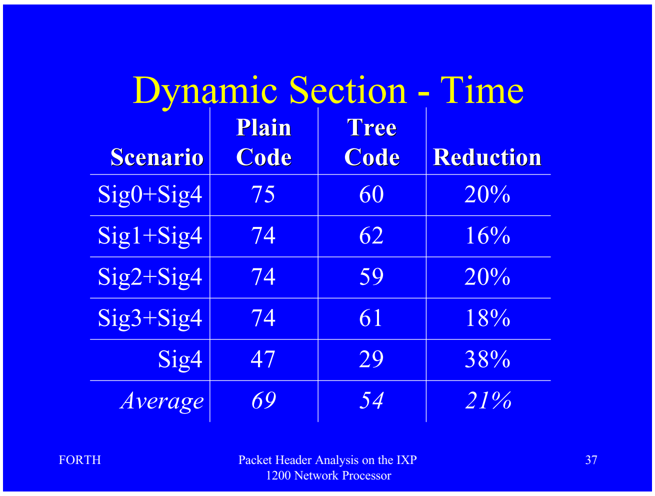
- Evaluate Static Section
  - Headroom
- Evaluate Dynamic Section
  - Space : How many signatures can fit
  - Time : How many signatures can be checked
- 100 Mbit/s

Static Section - Headroom



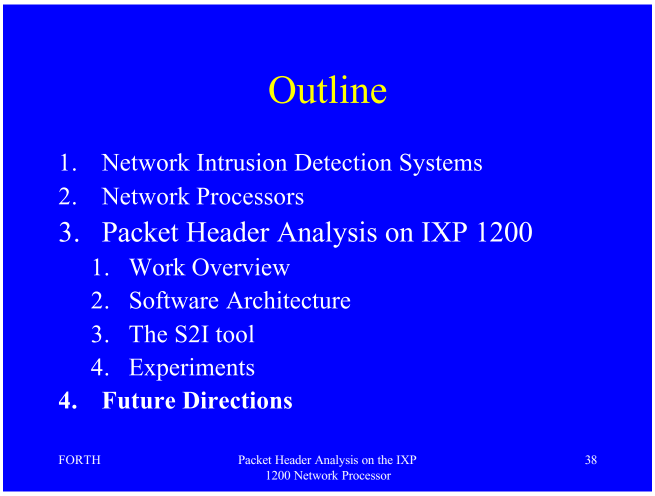
Dynamic Section - Space

Signature File	**No signatures**	**Plain code instr.**	**Tree code instr.**	**Reduction**
Icmp-info	79	>2000	479	>69%
Backdoor	44	1531	886	42%
Web-misc	18	401	277	31%
Virus	6	173	149	14%
Web.cgi	4	145	120	17%



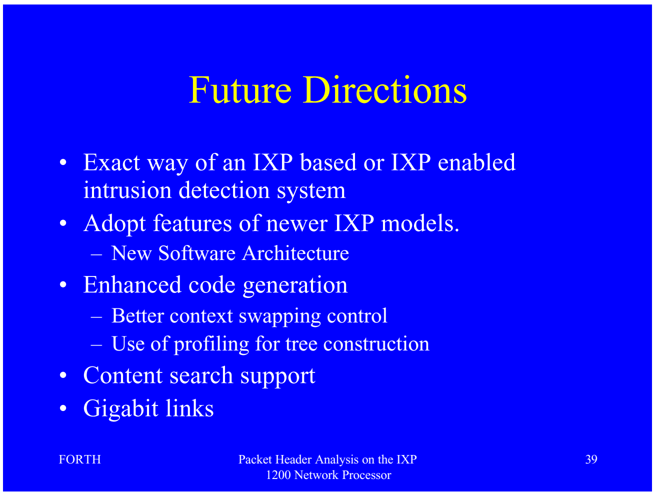
Dynamic Section - Time

	Plain	Tree	
Scenario	**Code**	**Code**	**Reduction**
Sig0+Sig4	75	60	20%
Sig1+Sig4	74	62	16%
Sig2+Sig4	74	59	20%
Sig3+Sig4	74	61	18%
Sig4	47	29	38%
Average	*69*	*54*	*21%*



Outline

1. Network Intrusion Detection Systems
2. Network Processors
3. Packet Header Analysis on IXP 1200
   1. Work Overview
   2. Software Architecture
   3. The S2I tool
   4. Experiments
4. **Future Directions**



Future Directions

- Exact way of an IXP based or IXP enabled intrusion detection system
- Adopt features of newer IXP models.
  - New Software Architecture
- Enhanced code generation
  - Better context swapping control
  - Use of profiling for tree construction
- Content search support
- Gigabit links