高速PCルータ研究 概観と論文紹介

小原 泰弘
NTTコミュニケーションズ
yasu@nttv6.jp

自己紹介

- WIDEプロジェクトメンバー
- GNU Zebra ospf6d 作者
- 慶應大学SFC村井研究室で博士取得
  - マルチパス経路計算アルゴリズム
- 北陸先端科学技術大学院大学(JAIST)助教
- カリフォルニア大学サンタクルーズ校(UCSC)ポスドク
  - 分散ストレージ・並列ファイルシステム研究
- 2013年からNTTコミュニケーションズ
- 高速PCルータ研究会(closed)主催

高速PCルータ研究会

- 2014/04 に始まったクローズドな研究会
- 楽しいことをやる・ゆるい対象領域/目的
- PCで大量のパケットを高速に処理する
  - 最低限の機能としてルータ、広い応用の可能性

- トラフィックジェネレータ
- トラフィックモニタ
- イーサネットブリッジ
- スイッチ
- 遅延エミュレータ
- トラフィックキャプチャマシン

- IDS/IPS/FW/DPI
- ロードバランサ
- ルートサーバ・リフレクタ
- ストレージノード
- ウェブサーバ
- PCクラスタ・超並列（スパコン）計算ノード

発表の目的

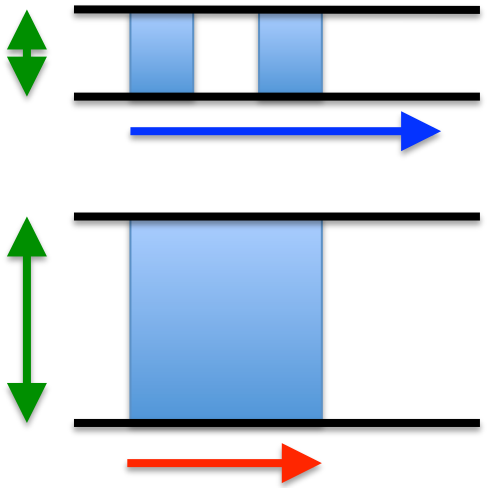
- 前提知識
- 歴史・過去の研究論文の紹介
- 何が難しいのか？
- 主なテクニック

単位の話

- ms: millisecond 1秒の 1/1,000
- us: microsecond 1秒の 1/1,000,000
  - 正しくはミューを使う。μs
  - u に似てるから代替する場合がある
  - usec ユーセック、などとも言う
- ns: nanosecond 1秒の 1/1,000,000,000
- ms > us > ns

スループットとレイテンシとpps

- スループット(throughput):
  - 単位時間あたりの処理量
  - 土管の太さ、水の量
  - bps: bits per second
- レイテンシ(latency):
  - 土管の長さ、蛇口ひねってから水出てくるまでの時間
  - ms / us / ns
- 高スループットルータ
- 低レイテンシルータ
- pps:
  - 単位時間あたりの処理回数
  - pps: packets per second



RIB v.s. FIB
Control Plane v.s. Forwarding Plane

- ルータでは Control Plane と Forwarding Plane の分離が進んだ
  - パケット転送（＝Forwarding Plane）は Interface Card / Interface Module で
  - 経路計算、経路制御プロトコルの動作（＝Control Plane）は CPU（Routing Module）で
- 経路は、計算して(=RIB)、Interface Card に転送(=FIB)
- RIB: Routing Information Base
- FIB: Forwarding Information Base

Routing v.s. Forwarding

- Routing とは
  - パケットを経路づけること
  - 経路を計算すること
  - 経路を知らせる（伝達する）こと
- Forwarding とは
  - パケットを決められた経路に転送すること
  - IP ヘッダ処理（TTL減算など）、データリンクアドレス解決を含む

- 高速PCルータ研究分野では、
  - 経路ルックアップするのがRoutingモード
  - 経路ルックアップしないのがForwardingモード

ルータの仕事

- NICからパケットを受信
  - スケジューリング
  - コピー
- IPヘッダを処理
- 終点で経路を検索
  - フルルート50万経路
- ネクストホップ向けNICにパケットを送信

パケットの大きさと pps

- 10.0 * 1000 / ((8 + 60 + 4 + 12) * 8)　(Mpps)
  - Preemble 8 byte
  - 最小ethernet frame 60 byte
  - FCS 4 byte
  - Interframe gap 12 byte
- = 14.88 Mpps
- 40.0 * 1000 / ((8 + 60 + 4 + 12) * 8) = 59.52 Mpps
- 100.0 * 1000 / ((8 + 60 + 4 + 12) * 8) = 148.80 Mpps

何が難しいのか

- 10Gbps 64B packet: 67.20 ns
  - 1000.0 * 1000 * 1000 / (14.88 * 1000 * 1000)
  - = 67.20
- 100Gbps 64B packet: 6.72 ns

- DDR3-800のCASレイテンシは 20ns
- DDR3-2000のCASレイテンシは 9ns

テクニック

- スケジューリング方法
  - Interrupt v.s. polling (i.e., spin-wait, busy-wait)
- 並列化
  - NICのマルチキュー、マルチコアCPU、GPU
- パイプライニング
- 無駄(overhead)を減らす
  - バッチ処理
  - コピー回数を減らす
  - バッファ管理

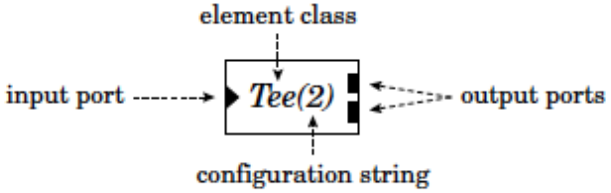
技術概要

- Click (SOSP ’99) (73Kpps)
- Click (TOCS ‘00) (452Kpps/333Kpps)
- NAPI (ALS ’01) (88Kpps) from 27Kpps
- RouteBricks (SOSP ‘09) (18.96Mpps/ 12(?)Mpps)
- PacketShader (SIGCOMM ’11)(58.4Mpps/50+ (?)Mpps)
- NetMap (USENIX ATC ’12) (14.88Mpps) at 1-core 900MHz

Click Modular Router (1)

- E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek., “The Click modular router”, ACM Transactions on Computer Systems 18(3), August 2000, pages 263-297.
- SOSP ‘99, TOCS ’00
- Still active
  - Used in SEATTLE(SIGCOMM’08)
  - Used in RouteBricks(SOSP’09)
  - Last release: 2011/09/24
  - Last commit: 2014/04/06
- Interrupt v.s. Polling (i.e., spin-wait, busy-wait)
- Pentium III 700 MHz, 100Mbps Ether DEC tulip PCI, …
- Linux 84kpps, Polling Linux 284kpps
- Routing 333kpps, Forwarding 452kpps
- Time taken: 2,798 ns/packet

Click Modular Router (2)



A sample element. Triangular ports are inputs and rectangular ports are

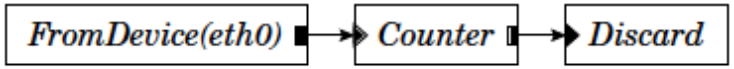


Fig. 2. A router configuration that throws away all packets.

```
// Declare three elements ...
src :: FromDevice(eth0);
ctr :: Counter;
sink :: Discard;
// ... and connect them together
src -> ctr;
ctr -> sink;

// Alternate definition using syntactic sugar
FromDevice(eth0) -> Counter -> Discard;
```

Fig. 6. Two Click-language definitions for the trivial router of Figure 2.

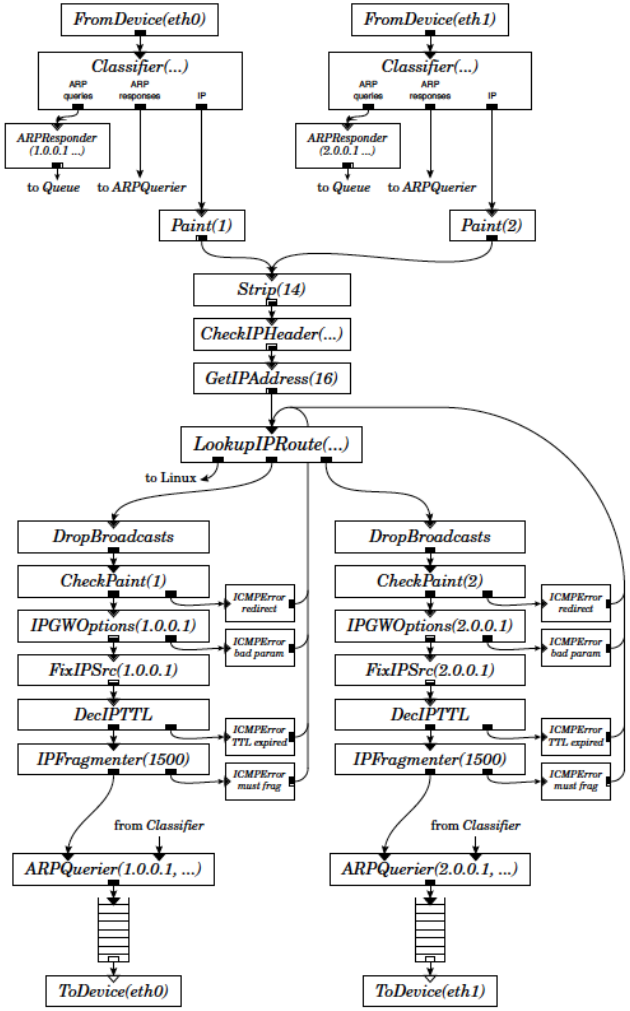


Fig. 8. An IP router configuration.

RouteBricks (1)

- M. Dobrescu, N. Egi, K. Argyraki, B.-G. Chun, K. Fall, Gianluca Iannaccone, Allan Knies, Maziar Manesh, Sylvia Ratnasamy, "RouteBricks: Exploiting Parallelism To Scale Software Routers, " 22nd ACM Symposium on Operating Systems Principles (SOSP), October 2009
- SOSP '09
- Click-base
- Leverages:
  - multi-cores.
  - multi-queue NICs.
  - batch processing.
- NUMA-aware data placement didn't make differences (in contrast to PacketShader).
- Achieved 18.96Mpps Forwarding, 12(?)Mpps Routing
- (Routing: 256K routes with memory expansion (DIR-24-8-BASIC))

RouteBricks (2)

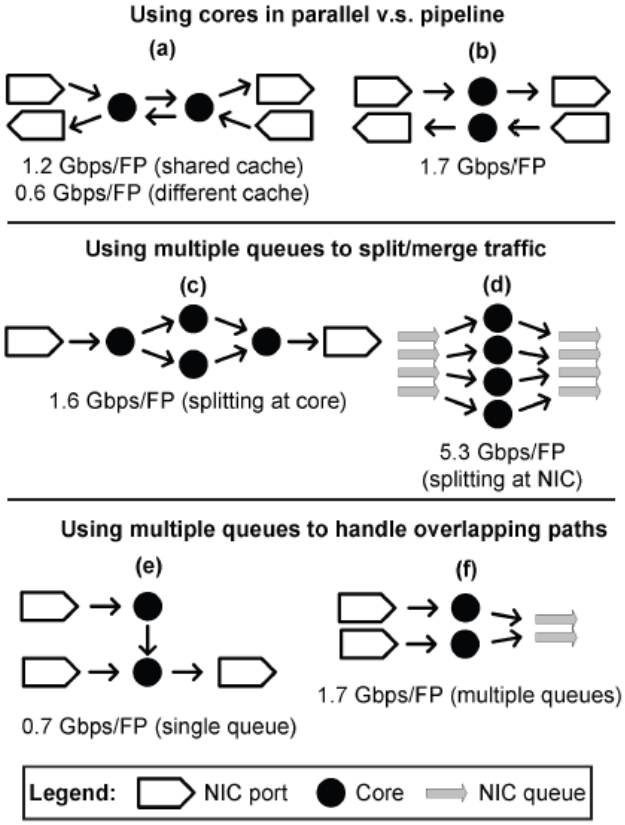


Figure 6: Forwarding rates with and without multiple queues.

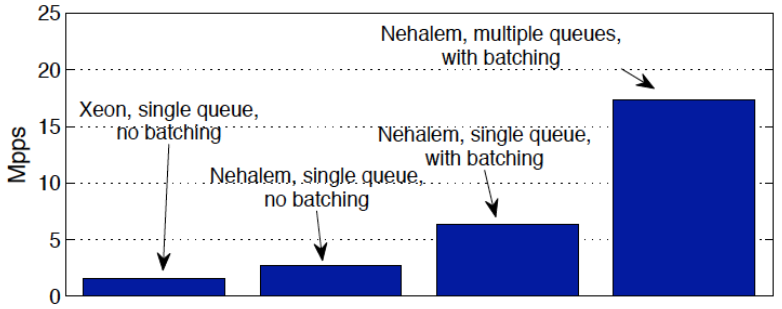
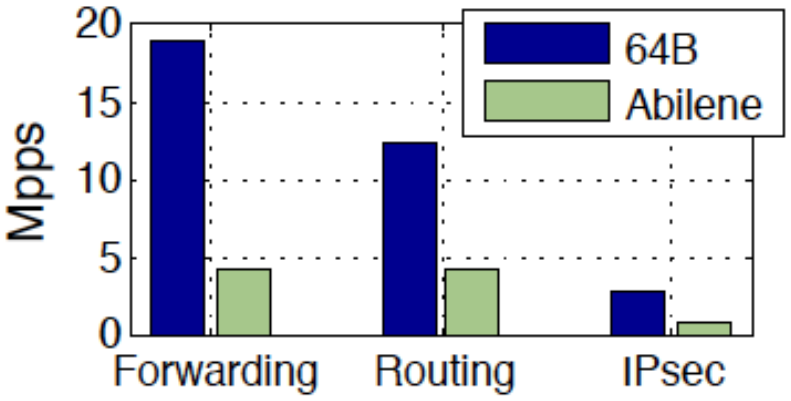


Figure 7: Aggregate impact on forwarding rate of new server architecture, multiple queues, and batching.



PacketShader (1)

- S. Han, K. Jang, K. Park and S. Moon. “PacketShader: a GPU-accelerated Software Router.” In proceedings of ACM SIGCOMM 2010, Delhi, India. September 2010.
- Speeds routing lookups (not just forwarding)
- Forwarding: 41.1 Gbps or 58.4 Mpps
- IPv4 Routing (DIR-24-8-BASIC, 282K routes): 39Gbps.
- IPv6 Routing (Waldvogel, 200K routes): 38Gbps.
- Up to 130us/200us (IPv4/IPv6) routing delay (260us/400us for roundtrip)

PacketShader (2)

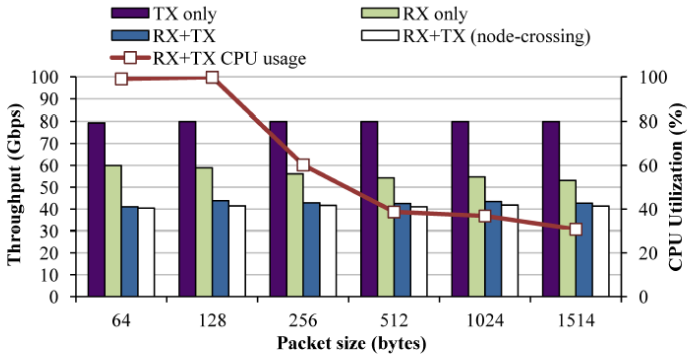


Figure 6: Performance of our packet I/O engine

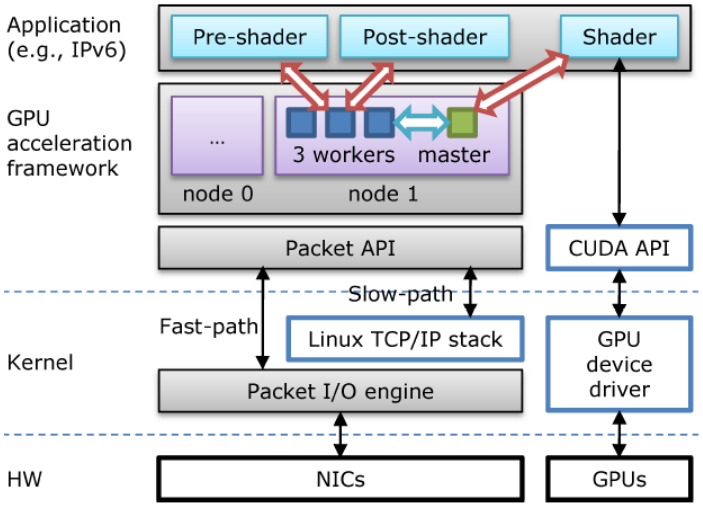
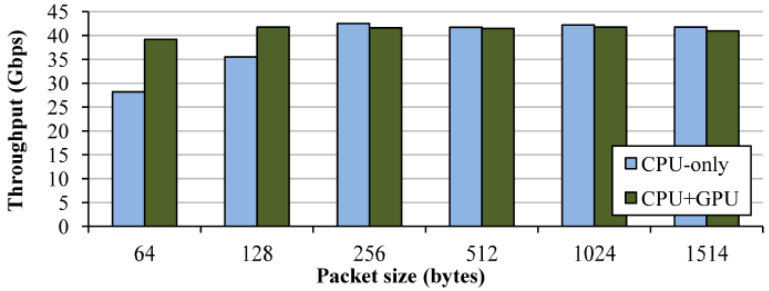
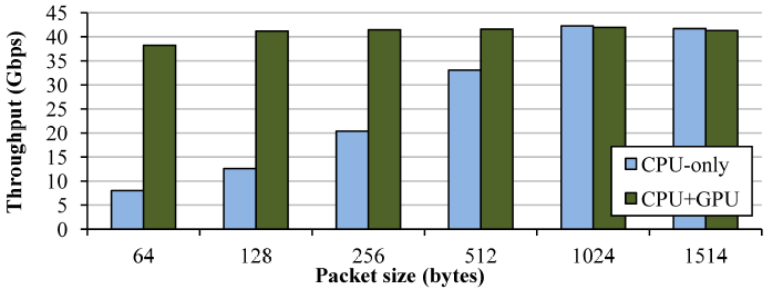


Figure 7: PacketShader software architecture

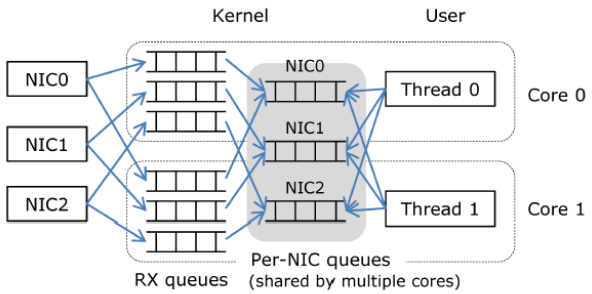


(a) IPv4 forwarding

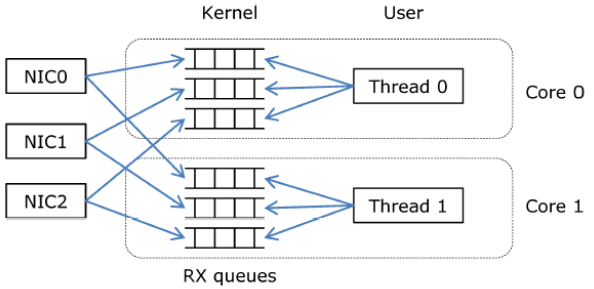


(b) IPv6 forwarding

PacketShader (3)



(a) Existing per-NIC queue scheme



(b) Our multiqueue-aware packet I/O scheme

Figure 8: User-level packet I/O interfaces

(a) Cores contend for queues.
Leads to expensive cache bouncing and lock contention.
(b) Better coupling of queues and cores.

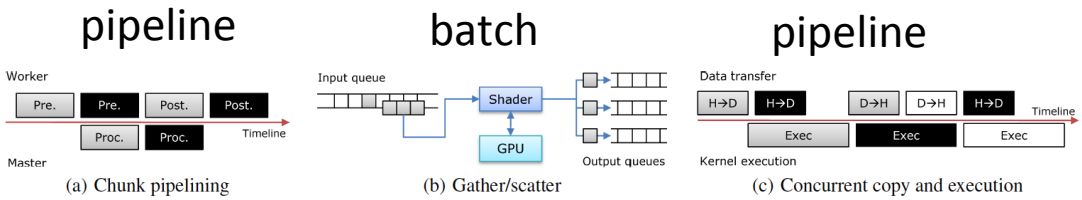


Figure 10: Optimizations on GPU acceleration

- Buffer management
- Pipelining
- Batch processing
- Decoupling among queues and cores

とはいえ、、、

- GDDR5？
- 480並列

NetMap (1)

- Luigi Rizzo, “netmap: a novel framework for fast packet I/O,” Usenix ATC'12, Boston, June 2012
- Forwarding 14.88 Mpps (1 core 900 MHz)
- Objective: to move packets quickly between the apps and the network cards.
- Reduced copy, modified buffer management.
- Protection. User-land v.s. Kernel.
- 408 ns / batch.
- Implementation on Linux/FreeBSD/Click.

NetMap (2)

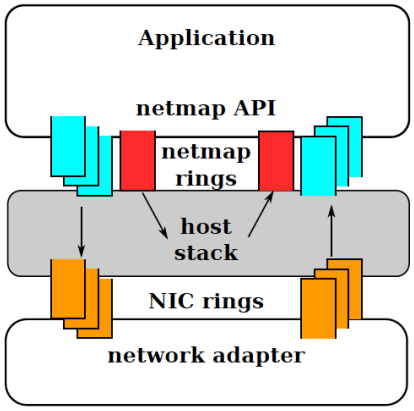


Figure 3: In netmap mode, the NIC rings are disconnected from the host network stack, and exchange packets through the netmap API. Two additional netmap rings let the application talk to the host stack.

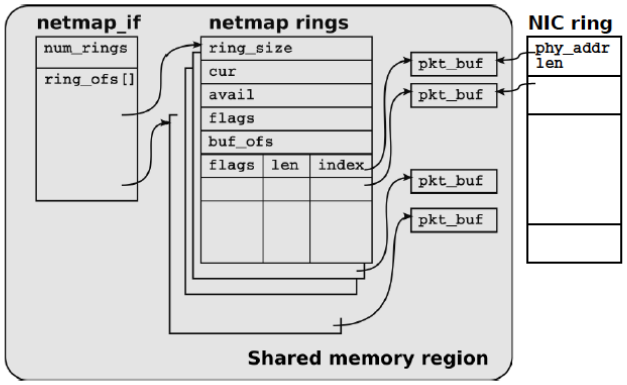


Figure 4: User view of the shared memory area exported by netmap.

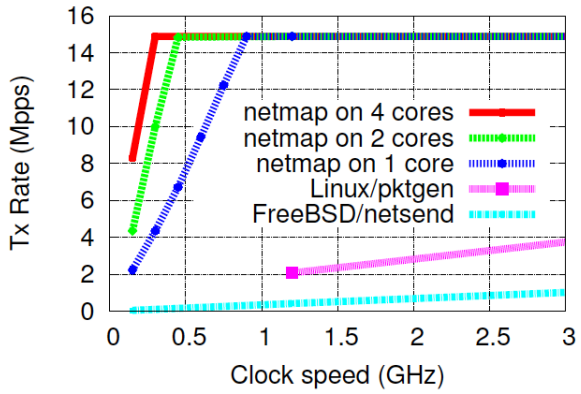


Figure 5: Netmap transmit performance with 64-byte packets, variable clock rates and number of cores, compared to `pktgen` (a specialised, in-kernel generator available on linux, peaking at about 4 Mpps) and a `netsend` (FreeBSD userspace, peaking at 1.05 Mpps).

Configuration	Mpps
netmap-fwd (1.733 GHz)	14.88
netmap-fwd + pcap	7.50
click-fwd + netmap	3.95
click-etherswitch + netmap	3.10
click-fwd + native pcap	0.49
openvswitch + netmap	3.00
openvswitch + native pcap	0.78
bsd-bridge	0.75

Figure 8: Forwarding performance of our test hardware with various software configurations.

経路ルックアップ技術

- Waldvogel (SIGCOMM’97)
- Lulea (SIGCOMM’97)
- DIR-24-8-BASIC (INFOCOM’98)
- PBF (SIGCOMM’03)
- TreeBitmap (CCR’04)
- DXR (CCR’12)
- GPU-Click (ANCS’13)
- GAMT (ANCS’13)
- SAIL (SIGCOMM’14)

テクニック

- TCAM
- FPGA
- GPU
- CPU
  - 小さくしてキャッシュに載せる
  - multibit trie
  - path-, level- compression
  - bit vector
  - leaf-pushing
  - prefix-length binary tree
  - Bloom filter

Waldvogel

- M. Waldvogel, G. Varghese, J. Turner, B. Plattner. “Scalable High-Speed IP Routing Lookups.” Proc. ACM SIGCOMM 1997 , pp. 25-36, Cannes, France.

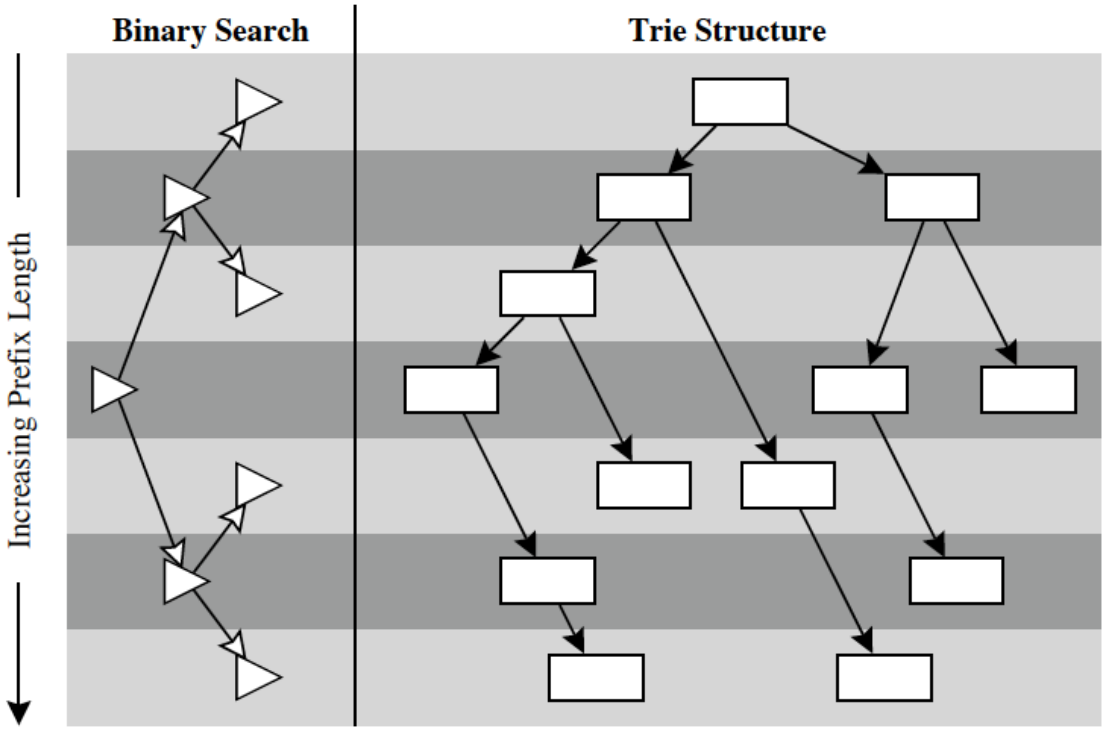


Figure 4: Binary Search on Trie Levels

Luleå

- スウェーデン　ルレオ（ルーレオー）大学
- M. Degermark, A. Brodnik, S. Carlsson, S. Pink, "Small Forwarding Tables for Fast Routing Lookups" Proc. ACM SIGCOMM `97

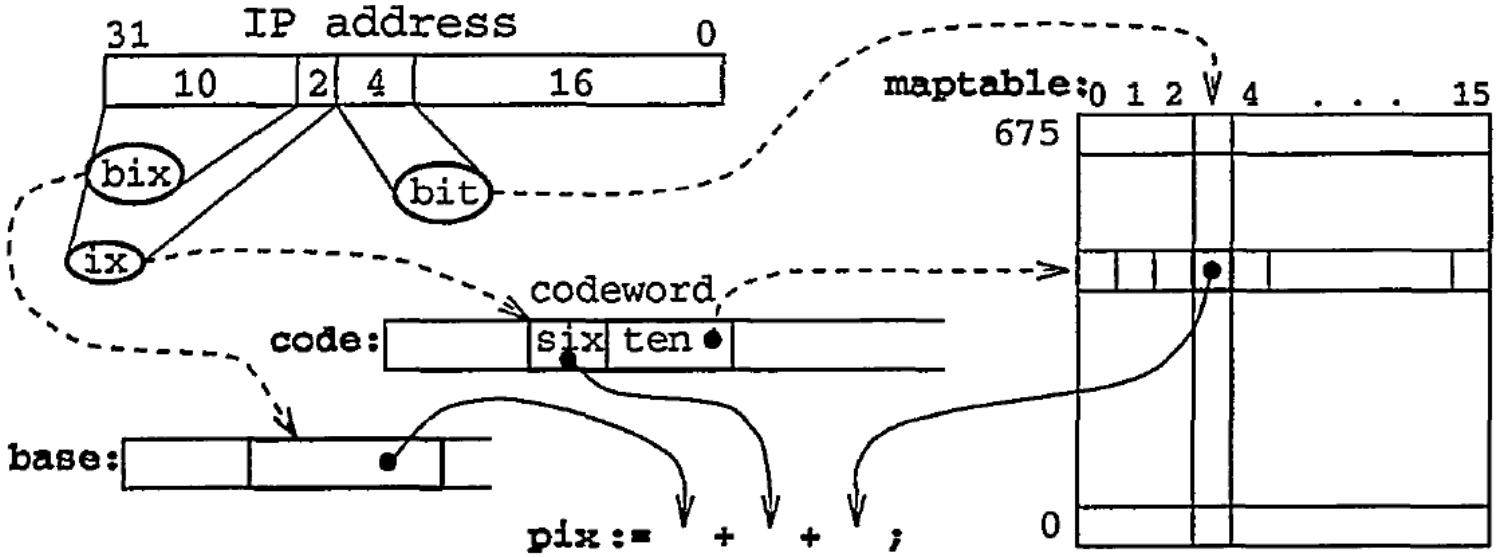


Figure 9: Finding the pointer index.

DIR-24-8-BASIC

- P. Gupta, S. Lin, and N. McKeown. “Routing Lookups in Hardware at Memory Access Speeds.” In Proceedings of the IEEE INFOCOM Conference, San Francisco, CA, USA, March 1998.
- INFOCOM ‘98

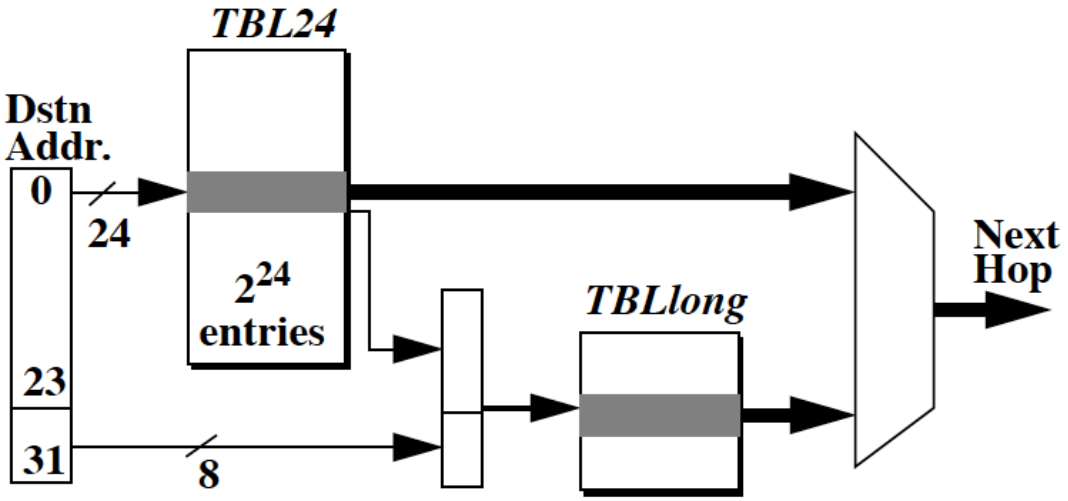


Figure 2: Proposed *DIR-24-8-BASIC* architecture. The next hop result comes from either *TBL24* or *TBLlong*.

SAIL (1)

- T. Yang, G. Xie, Y. B. Li, Q. Fu, A. X. Liu, Q. Li, L. Mathy, “Guarantee IP Lookup Performance with FIB Explosion,” ACM SIGCOMM ‘14.
- FPGA, CPU, GPU implementations
- Lookup: 673.22~708.71 Mpps for real traffic
- Lookup: 231.47~236.08 Mpps for random
- IP address locality

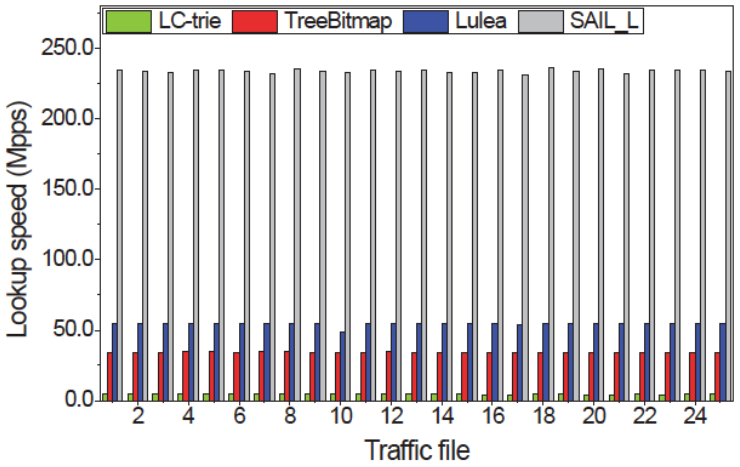


Figure 10: Lookup speed with random traffic on FIB_{2013}.

SAIL (2)

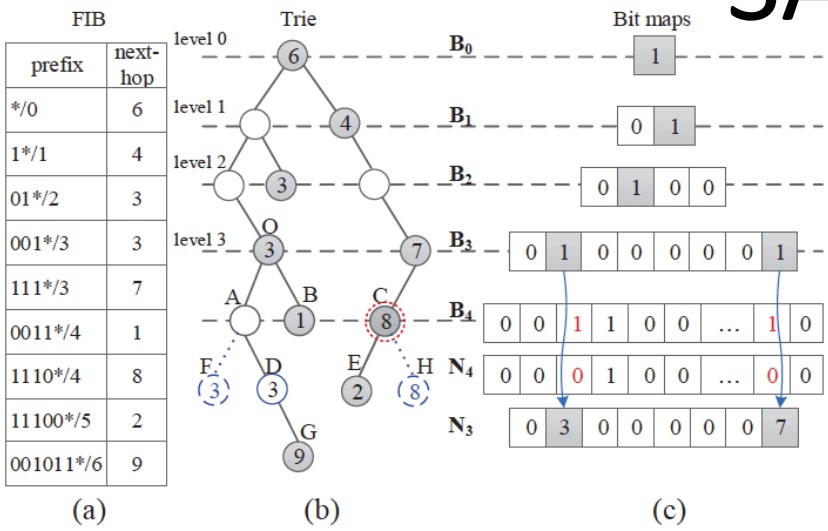


Figure 2: Basic SAIL Algorithm.

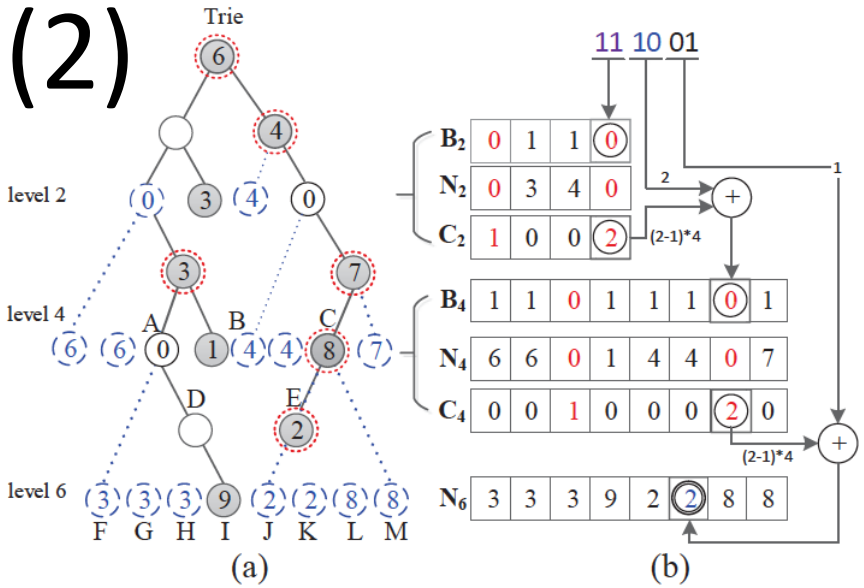


Figure 3: Example data structure of SAIL_L.

Algorithm 2: SAIL_L

Input: Bit map arrays: B_{16}, B_{24}
Input: Next hop arrays: N_{16}, N_{24}, N_{32}
Input: Chunk ID arrays: C_{16}, C_{24}
Input: a: an IP address
Output: the next hop of the longest matched prefix.

```
1 if B16[a >> 16]=1 then
2 |   return N16[a >> 16]
3 end
4 else if
  B24[(C16[a >> 16] − 1) << 8 + (a << 16 >> 24)] then
5 |   return
  |   N24[(C16[a >> 16] − 1) << 8 + (a << 16 >> 24)]
6 end
7 else
8 |   return N32[(C24[a >> 8] − 1) << 8 + (a&255)]
9 end
```

	Finding prefix length	Finding next hop
Prefix length 0~24	B_{16} , C_{16}, B_{24}	N_{16}, N_{24}
Prefix length 25~32	C_{24}	N_{32}

Figure 4: Memory management for SAIL_L.

まとめ

- 高速PCルータ技術
  - Click (SOSP ’99) (452Kpps/333Kpps)
  - NAPI (ALS ’01) (88Kpps) from 27Kpps
  - RouteBricks (SOSP ‘09) (18.96Mpps/12(?)Mpps)
  - PacketShader (SIGCOMM ’11)(58.4Mpps/50+(?)Mpps)
  - NetMap (USENIX ATC ’12) (14.88Mpps) at 1-core 900MHz
- 経路検索技術
  - Waldvogel (SIGCOMM’97)
  - Lulea (SIGCOMM’97)
  - DIR-24-8-BASIC (INFOCOM’98)
  - PBF (SIGCOMM’03)
  - TreeBitmap (CCR’04)
  - DXR (CCR’12)
  - GPU-Click (ANCS’13)
  - GAMT (ANCS’13)
  - SAIL (SIGCOMM’14)