路由器设计与实现

中国科学院大学
University of Chinese Academy of Sciences

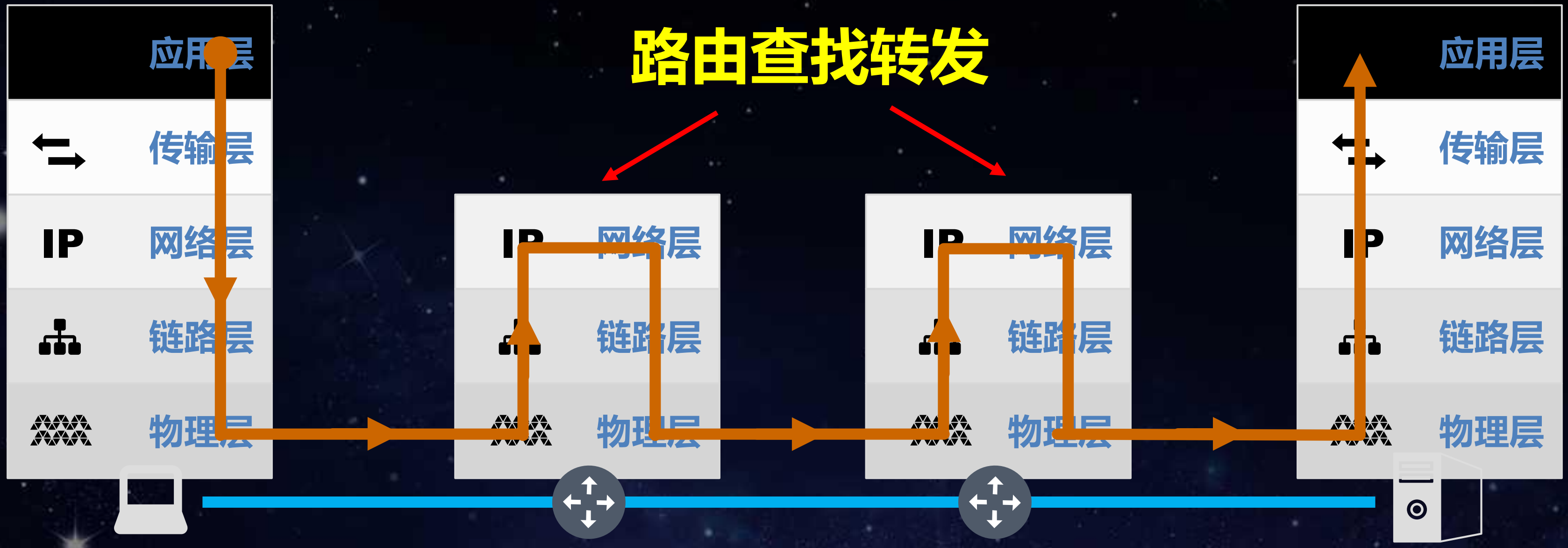
提纲

- 路由器概况
- 路由查找算法
- 队列管理
- 路由安全






路由器：互联网基础设备



延迟：传播、传输、查找转发

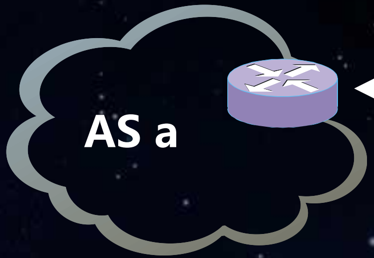
32 bits

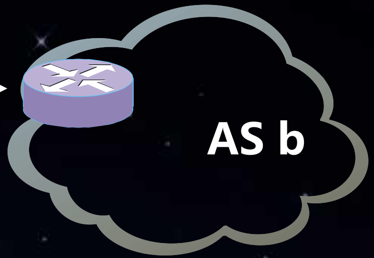
version (4 bits)	header length	DS	ECN	Total Length (in bytes) (16 bits)	
Identification (16 bits)				flags (3 bits)	Fragment Offset (13 bits)
TTL Time-to-Live (8 bits)		Protocol (8 bits)		Header Checksum (16 bits)	
Source IP address (32 bits)					
Destination IP address (32 bits)					

Ethernet Header	**IP Header**	TCP Header	Application data	Ethernet Trailer

Ethernet frame

路由信息交换：BGP例子





①建立TCP连接

②OPEN：参数协商AS ID、Hold time, router ID

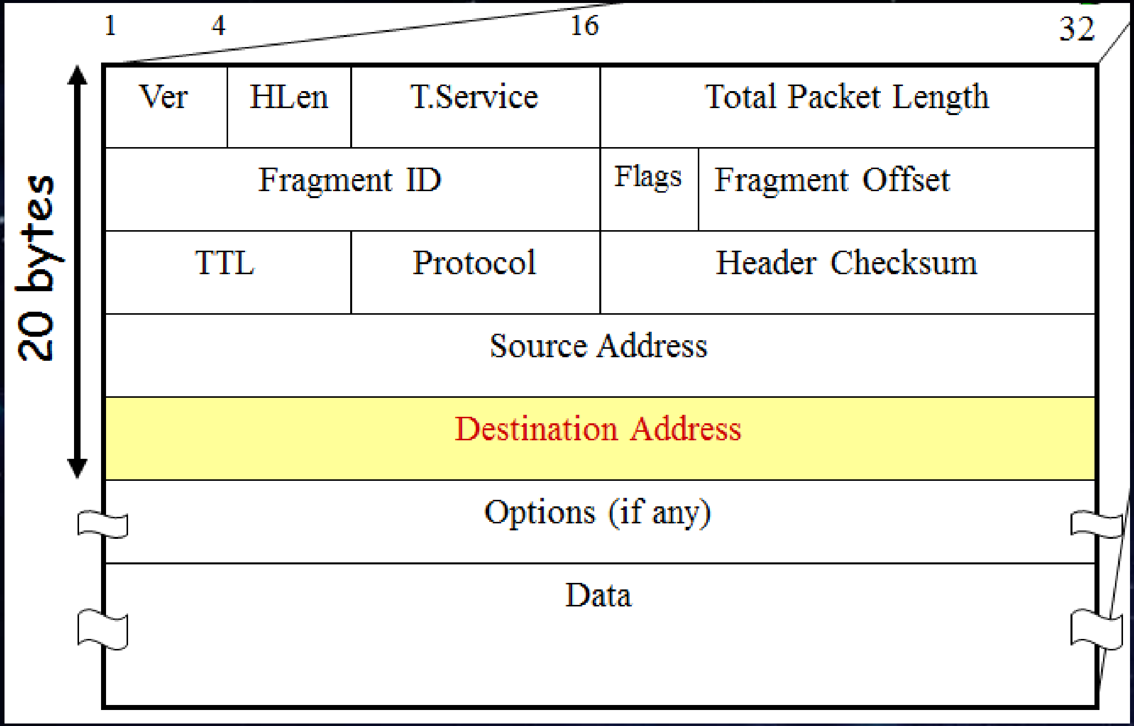
③Keepalive：完成对等建立，定期发送保持连接

④update：通告路由

Notification：报告错误，中止对等关系

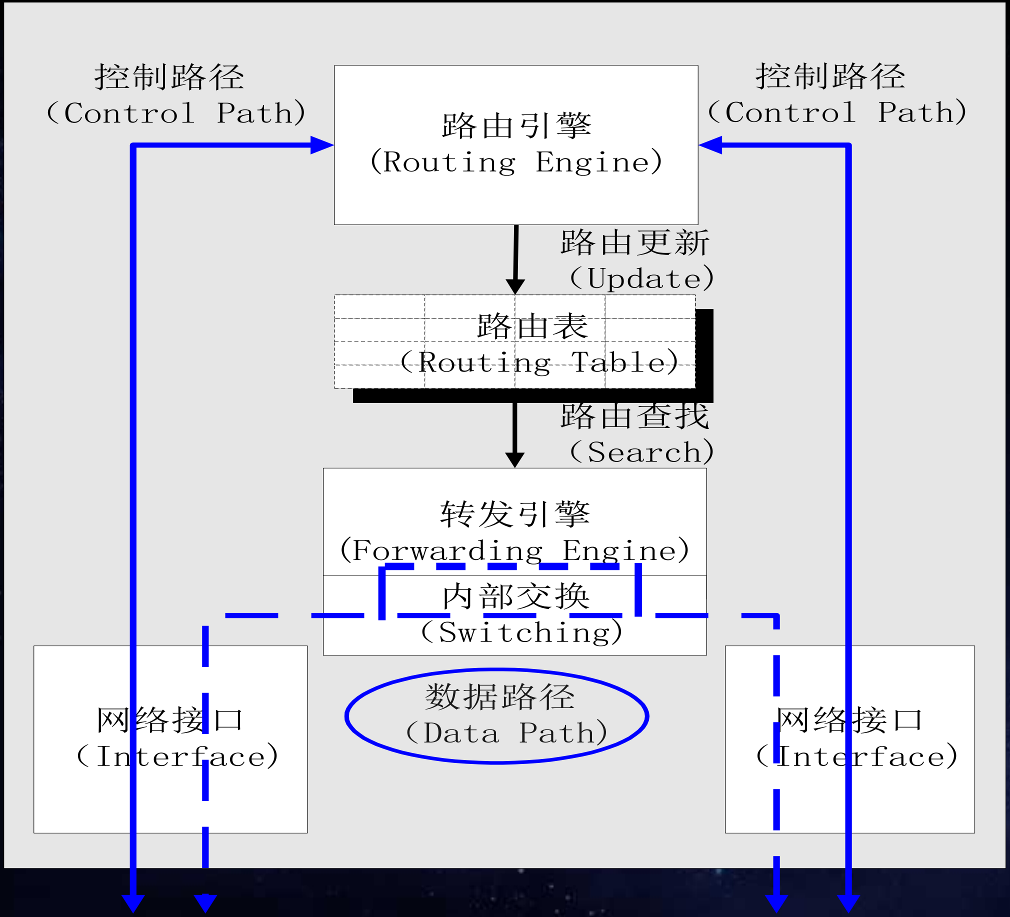
Route_refresh：路由策略变化，触发请求重新通告路由

路由查找转发



前缀	下一跳
0*	A1
1*	A2
00*	A3
11*	A4
010*	A5
111*	A6

路由器逻辑结构

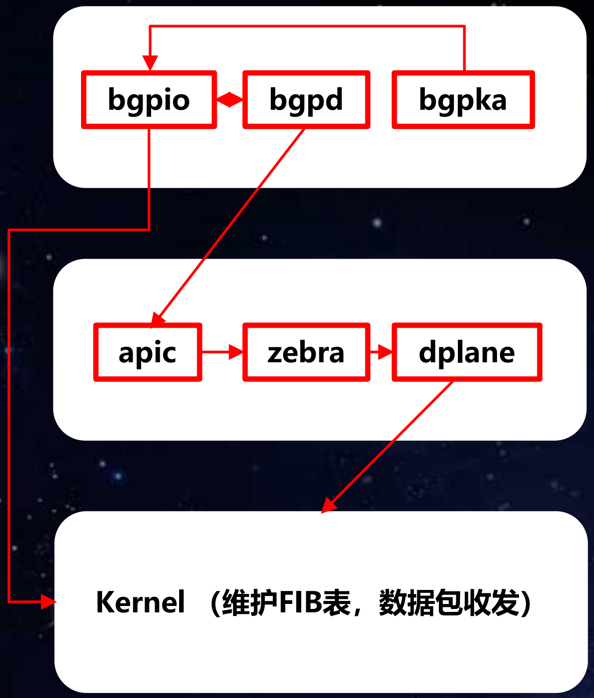


- 控制平面：主控卡
  - 路由协议与路由计算(路由表)
  - 快速收敛、最优路径、流量工程
- 数据平面：线卡
  - 数据包查找转发
  - 转发速度
  - 更新速度
  - QoS保障（队列）
- 管理平面：主控卡
  - 监测、配置

中国科学院大学 University of Chinese Academy of Sciences

实现实例：FRRouting BGP模块

(BGP、OSPF、RIP等路由协议守护进程)

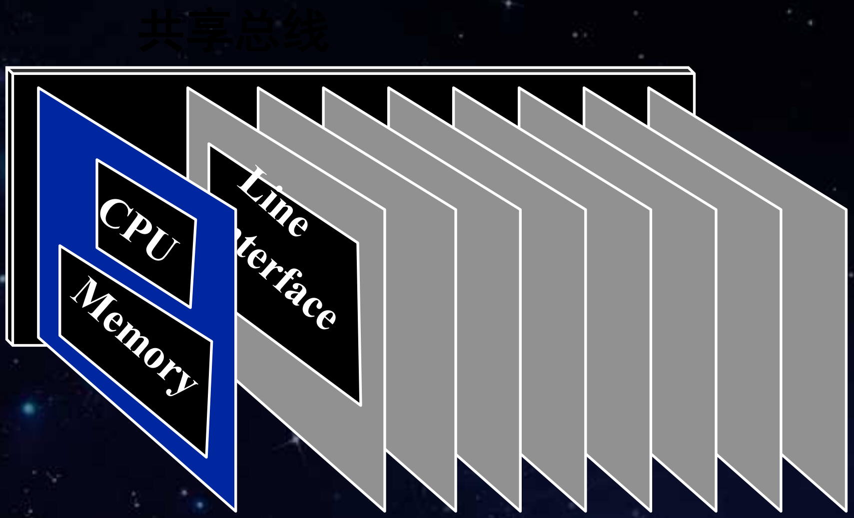


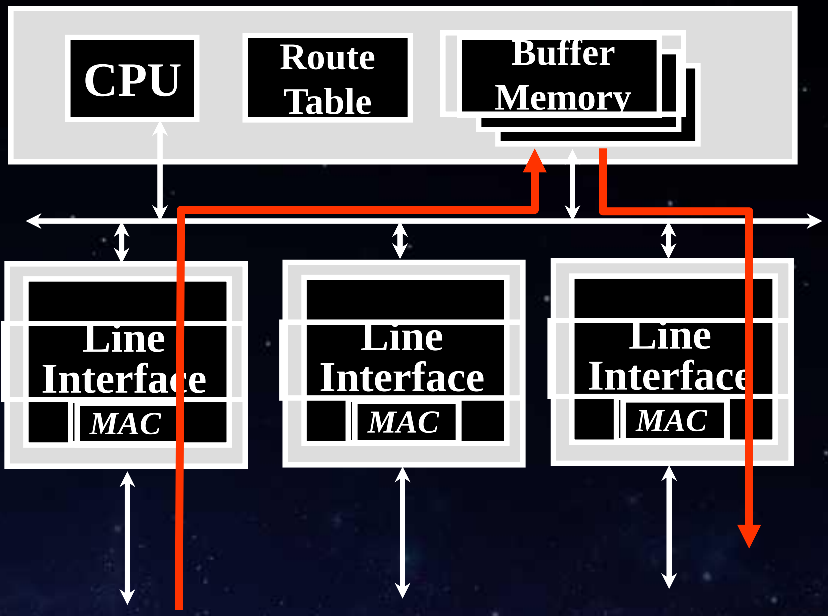
bgp
- bgpka：处理keepalive消息
- bgpio：收发包，报文入队唤醒bgpd处理
- bgpd：主线程，处理BGP业务

zebra
- apic：接受bgp进程路径优选后的消息
- dplane：将fib更新写入内核
- zebra：主线程，负责rib维护

数据包路由查找转发

第一代路由器：总线与单CPU

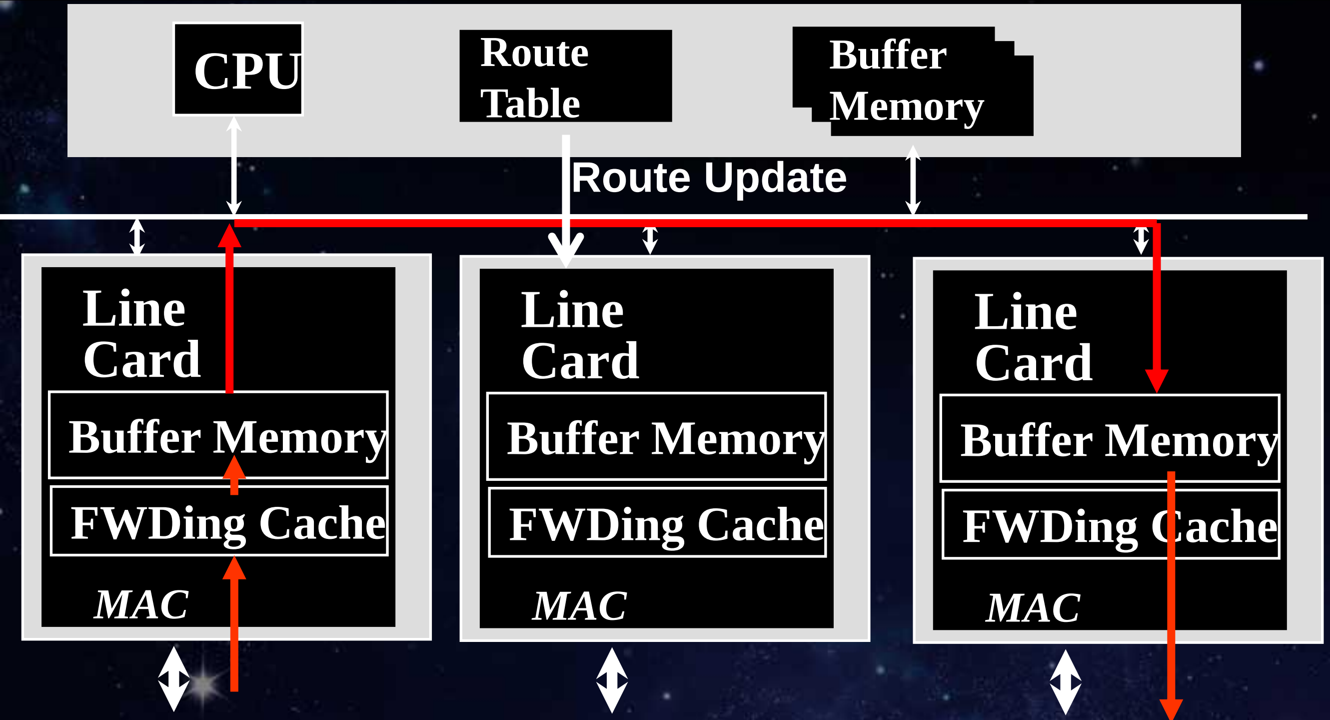




- 处理瓶颈：处理器查找转发数据包
  - 算法：Tree
- I/O瓶颈：数据包经过总线2次

From Prof. Nick MeKeown of Stanford

第二代路由器：总线与缓存转发



From Prof. Nick MeKeown of Stanford

- 线卡直接包转发
  - TCAM加速：路由表规模
  - 总线负载
  - 总CPU负载
- 问题
  - 路由cache命中率
  - 主路由表潜在瓶颈
  - 共享总线仍是瓶颈

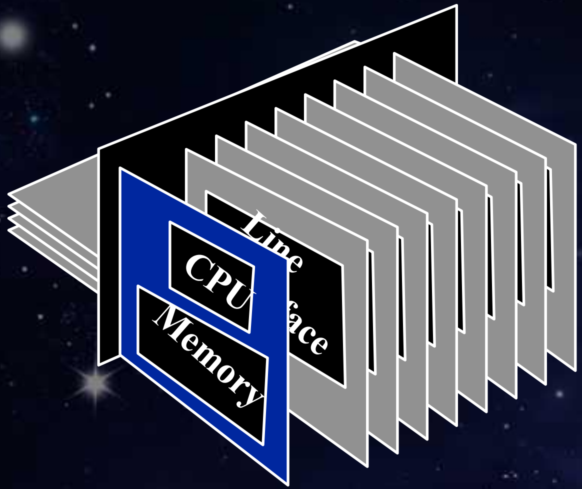
第三代路由器：交换背板与多转发引擎

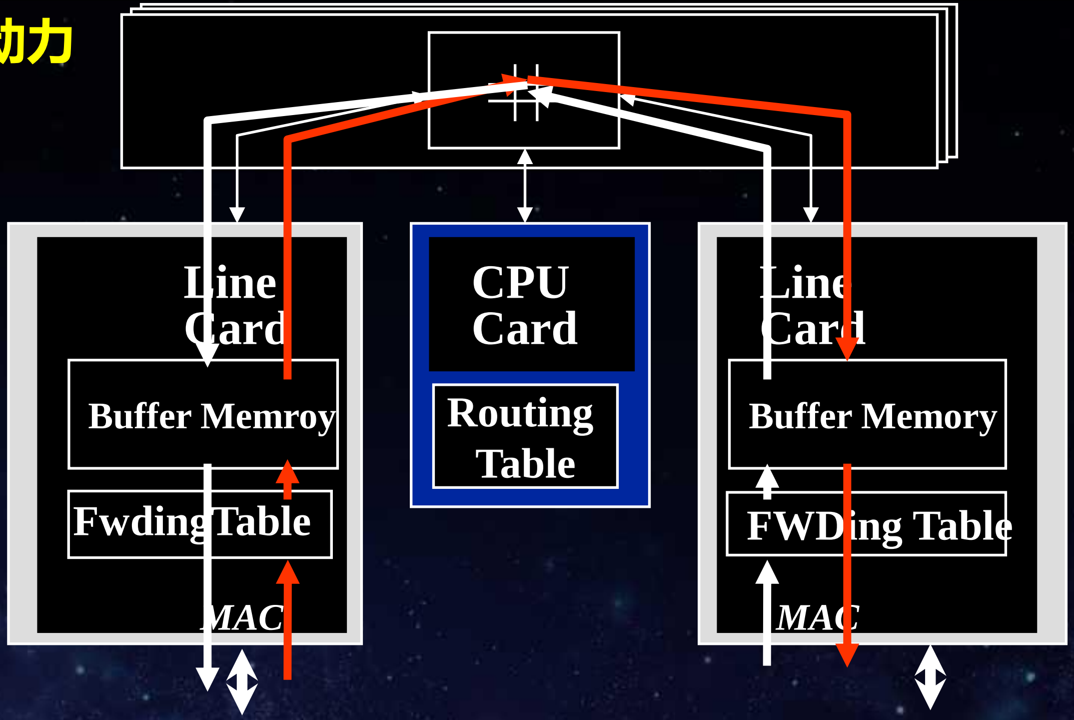
技术(器件)进步与流量需求是变革推动力

无阻塞交换背板(Switch Fabric)

FIB代替Cache

NP、ASIC





From Prof. Nick MeKeown of Stanford

路由查找：最长前缀匹配

DIP：110…

DIP：1110…

DIP：011…

DIP：000…

前缀	下一跳
0*	A1
1*	A2
00*	A3
11*	A4
010*	A5
111*	A6

A4 DIP：110…

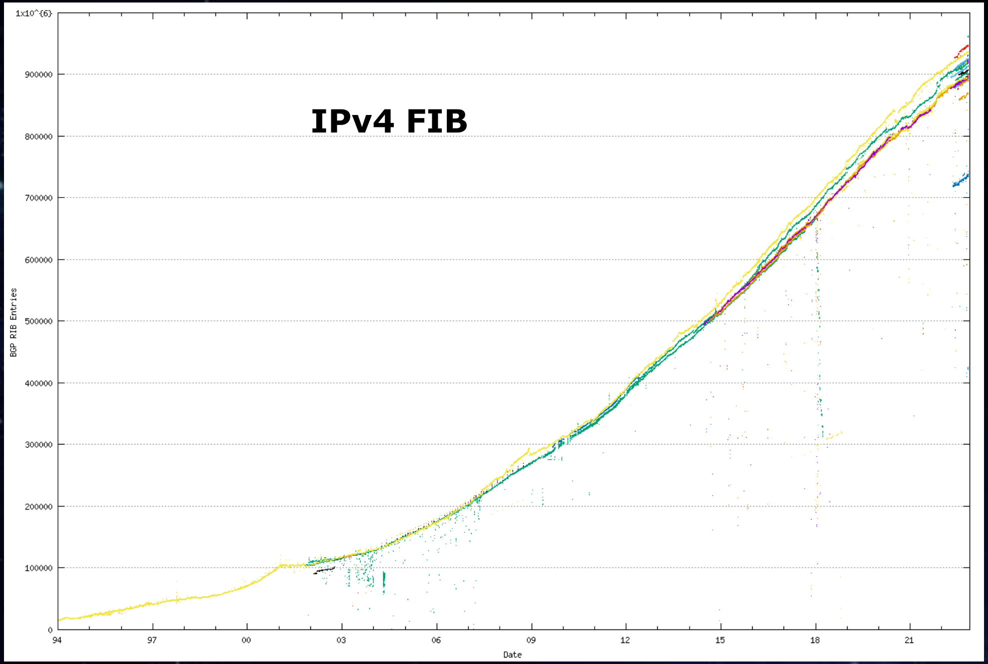
A6 DIP：1110…

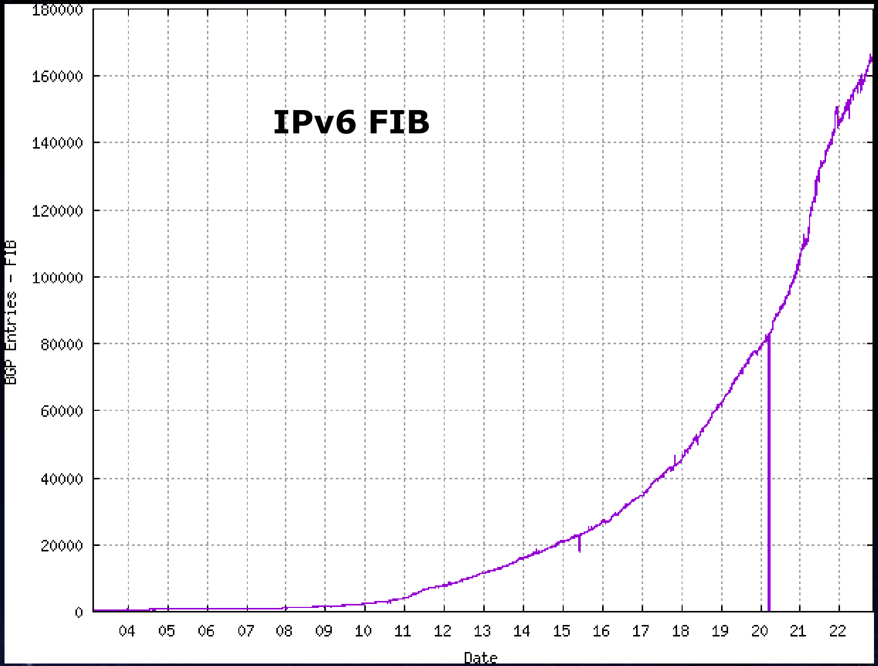
A1 DIP：011…

A3 DIP：000…

路由表项增长

http://www.potaroo.net，2022/11/6



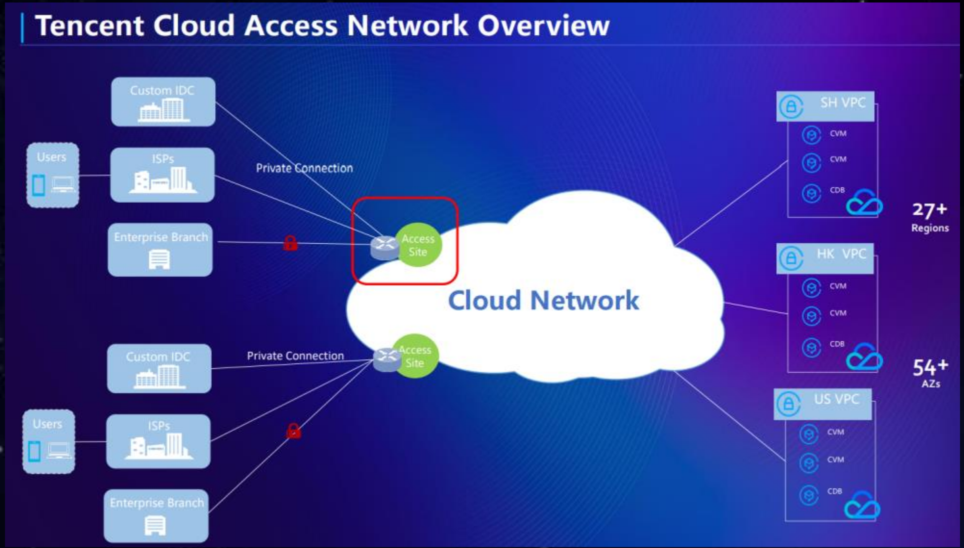


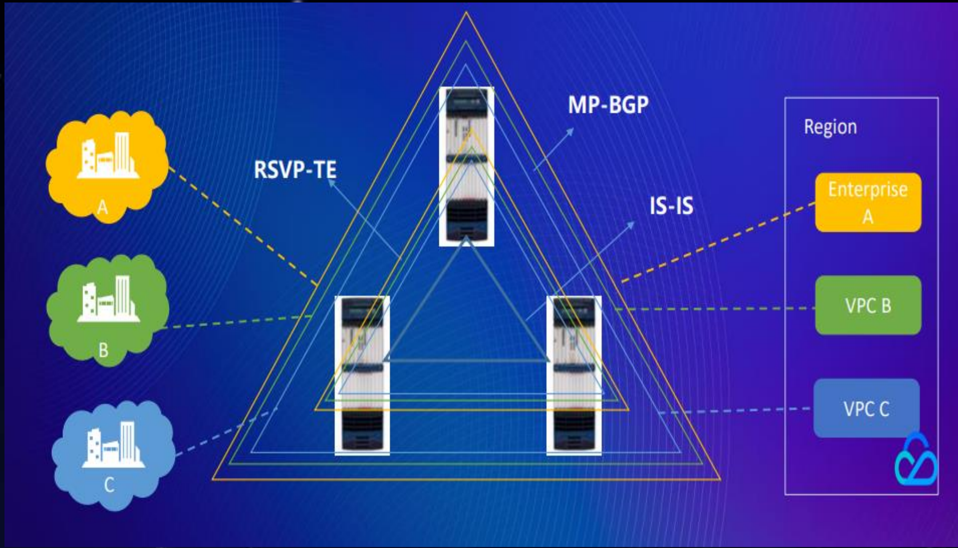
FIB表项增加

- AS路径平均长度基本稳定
- Tier1连接优势地位

- **IP查找性能需求**
  - 100GE (~150 Mpps，~6.7ns/packet)
  - 线卡处理速度~100Gbps * n

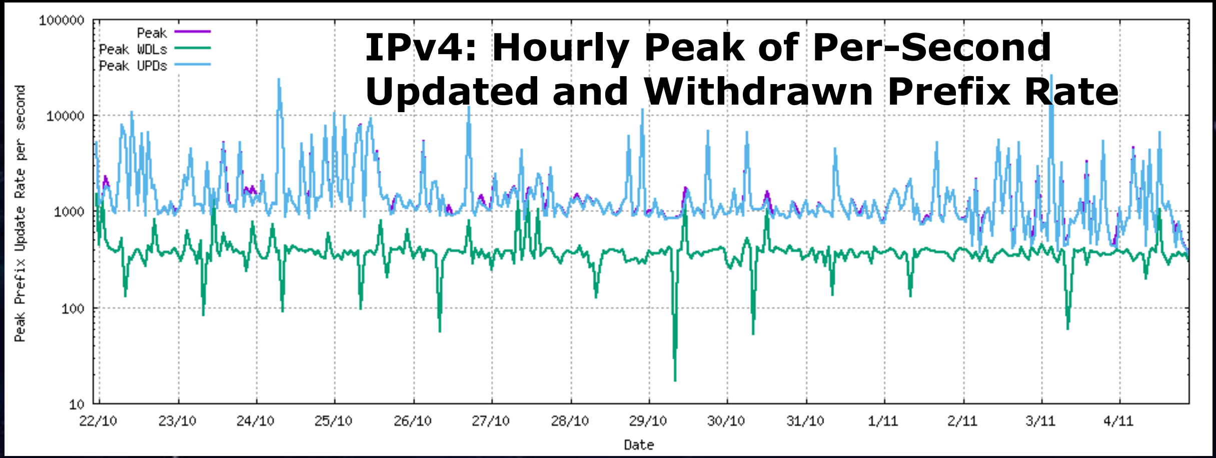
路由表项增长

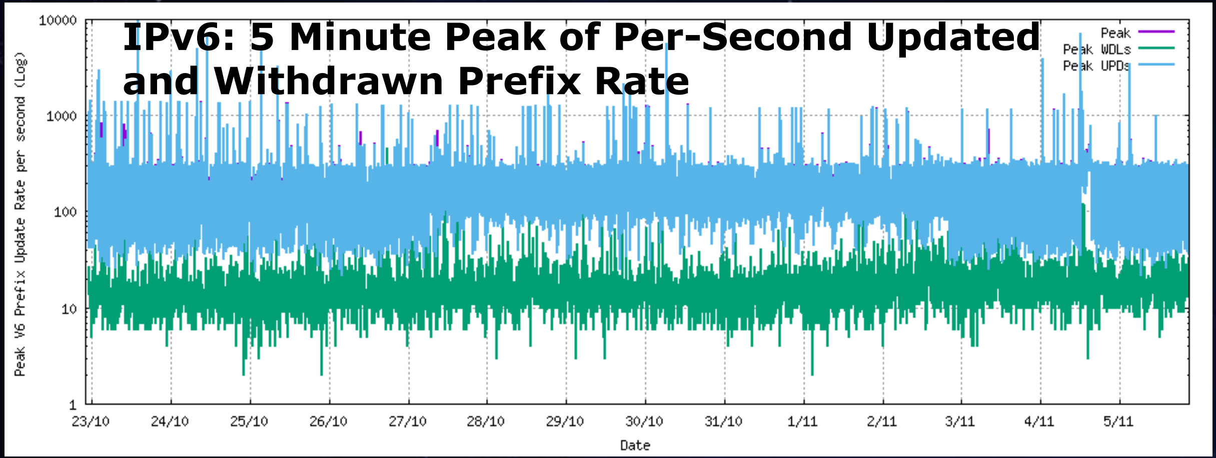




https://www.usenix.org/system/files/nsdi21_slides_shao.pdf

路由表更新



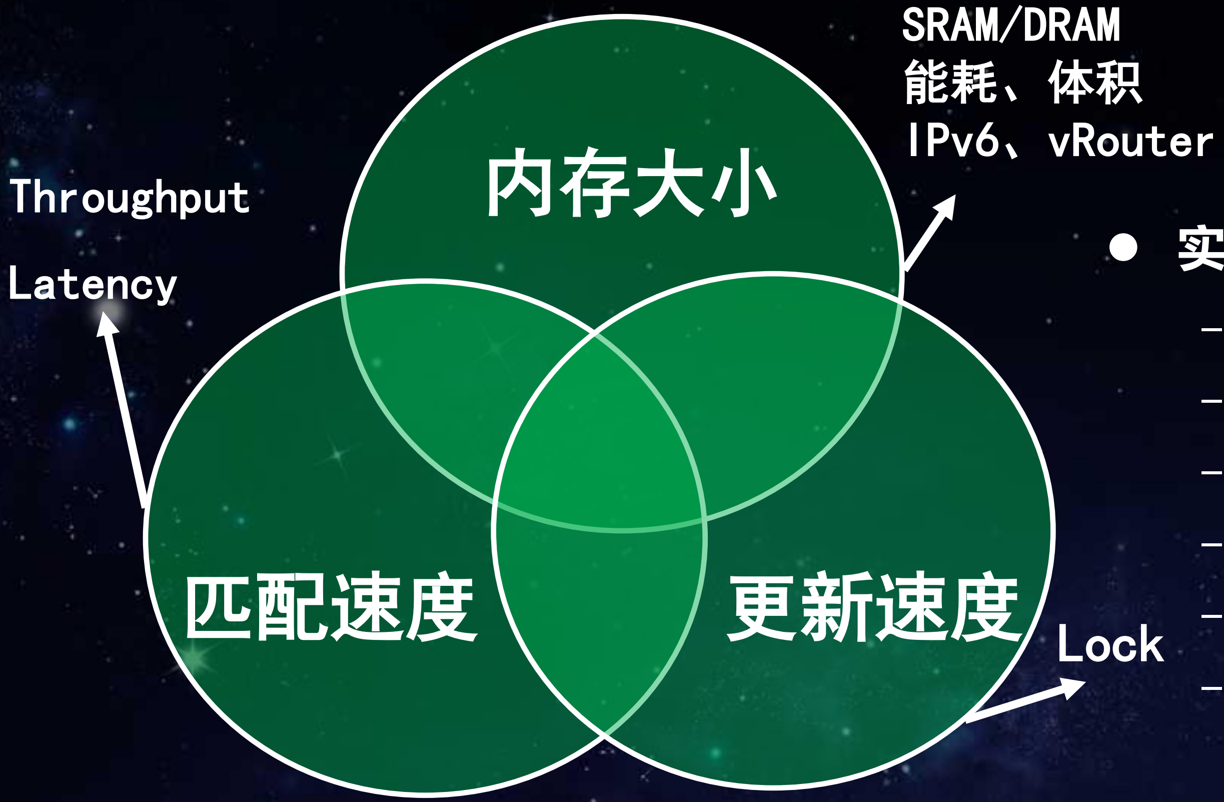


- 更新速率
  - BGP路由器峰值更新频率 > 1,000ups

http://www.potaroo.net，2022/11/6

University of Chinese Academy of Sciences

问题：如何设计最长前缀匹配算法?



- 实现硬件
  - CPU
  - TCAM
  - FPGA
  - NP
  - ASIC
  - GPU
- 算法
  - 软件实现算法
  - TCAM方法
  - FPGA方法
  - TCAM+FPGA

部分路由查找算法

- 路由查找
  - 算法
    - Trie
      - *Lulea (SIGCOMM'97)*
      - *LC-trie (JSAC'99)*
      - *TrieBitmap (SIGCOMM'04)*
      - *SAIL (SIGCOMM'14)*
      - *Poptrie (SIGCOMM'15)*
      - *XorOffsetTrie (ICC'22)*
      - *TrieMerg(Infocom'13)*
    - Hash
      - *Dir-24-8 (INFOCOM'98)*
      - *Flashtrie (INFOCOM'10)*
      - *Hi-BST (GLOBECOM'18)*
    - Binary search
      - *DXR (SIGCOMM CCR'12)*
  - 硬件实现
    - TCAM
      - *DIRPE(SIGCOMM'05)*
      - *Partial (ToN'18)*
      - *TCAM-vRouter(CoNEXT'12)*
    - GPU
      - *PacketShader (SIGCOMM'11)*
    - FPGA
      - *RCST (ANCS'05)*
      - *Elevator (ToN'05)*
      - *Hybrid (ToN'14)*

中国科学院大学
University of Chinese Academy of Sciences

顺序查找

前缀	下一跳
0*	A1
1*	A2
00*	A3
11*	A4
010*	A5
111*	A6

FIB：~900k表项

- 时间复杂度O(n)
- 空间复杂度O(n)
- 更新时间
  - Insert: O(1)
  - Del./Update: O(n)
- 优化：前缀长度倒序
  - 更新开销：路由表项移动

Hash查找

前缀	下一跳
0*	A1
1*	A2
00*	A3
11*	A4
010*	A5
111*	A6

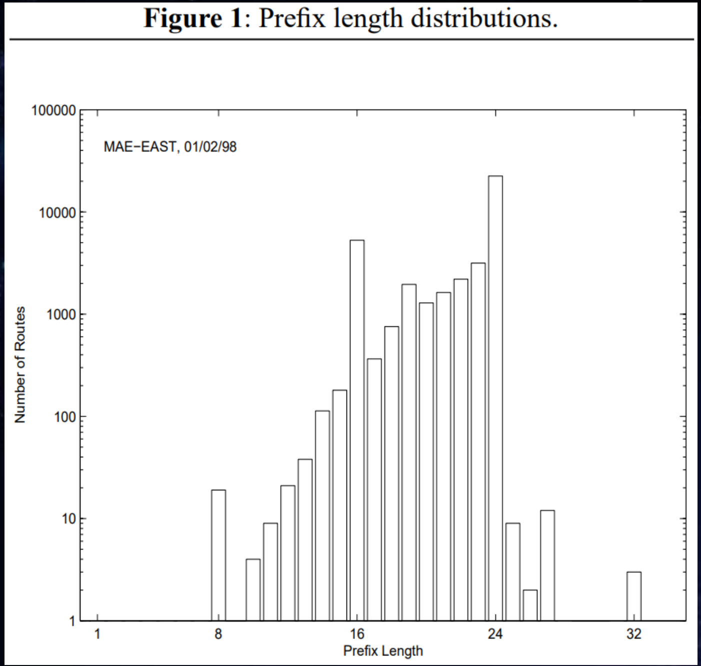
Keys	NH
010	A5
111	A6

Keys	NH
00	A3
11	A4

Keys	NH
0	A1
1	A2

- 按前缀长度倒序查找
- 并行化提升查找性能
- 哈希合并与链表

DIR-24-8



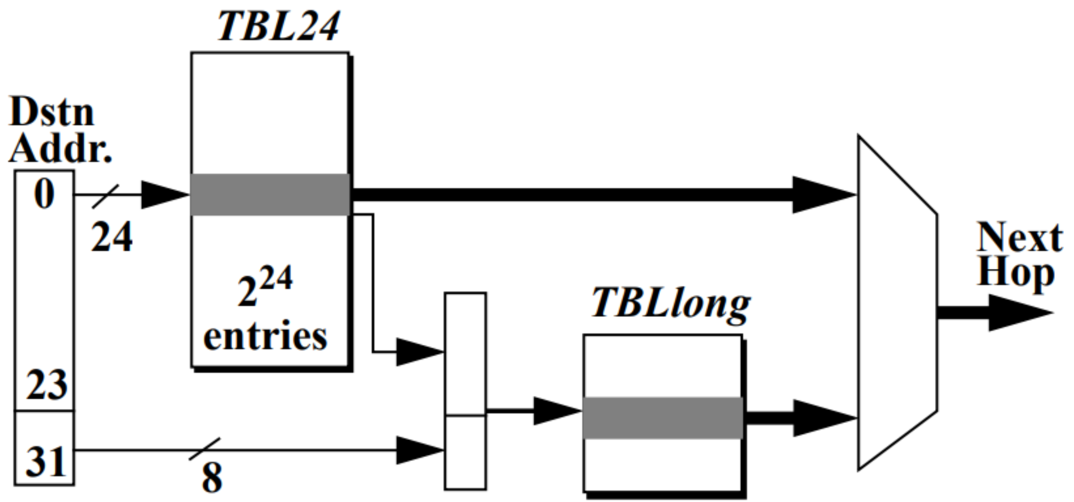


Figure 2: Proposed *DIR-24-8-BASIC* architecture. The next hop result comes from either *TBL24* or *TBLlong*.

Routing Lookups in Hardware at Memory Access Speeds, Infocom'98

DIR-24-8（续）

Key to table entries:

A = 10.54/16
B = 10.54.34/24
C = 10.54.34.192./26

Figure 3: *TBL24* entry format

If longest route with this 24-bit prefix is < 25 bits long:

0	Next Hop
1 bit	15 bits

If longest route with this 24 bits prefix is > 24 bits long:

1	Index into 2nd table
1 bit	15 bits

***TBL24*:**

Entry Number:		Contents:
⋮		⋮
10.53.255	–	—
10.54.0	0	A
10.54.1	0	A
⋮		⋮
10.54.33	0	A
10.54.34	1	123
10.54.35	0	A
⋮		⋮
10.54.255	0	A
10.55.0	–	—
⋮		⋮

***TBLlong*:**

Entry Number:	Contents:
⋮	⋮
123*256	B
123*256+1	B
123*256+2	B
⋮	⋮
123*256+191	B
123*256+192	C
123*256+193	C
⋮	⋮
123*256+255	C
124*256	C
⋮	⋮

256 entries allocated to 10.54.34 prefix

Figure 4: Example of two tables containing three routes.

前缀：一维空间区间投影

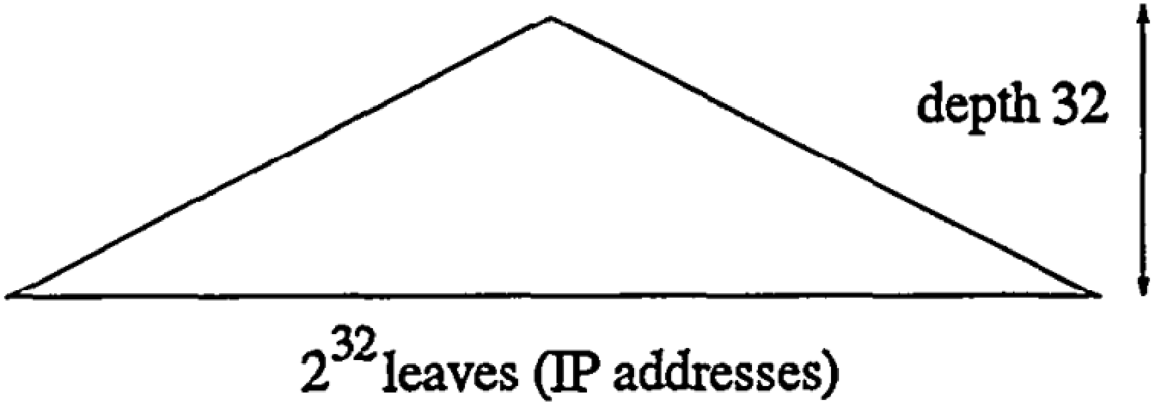


Figure 3: Binary tree spanning the entire IP address space.

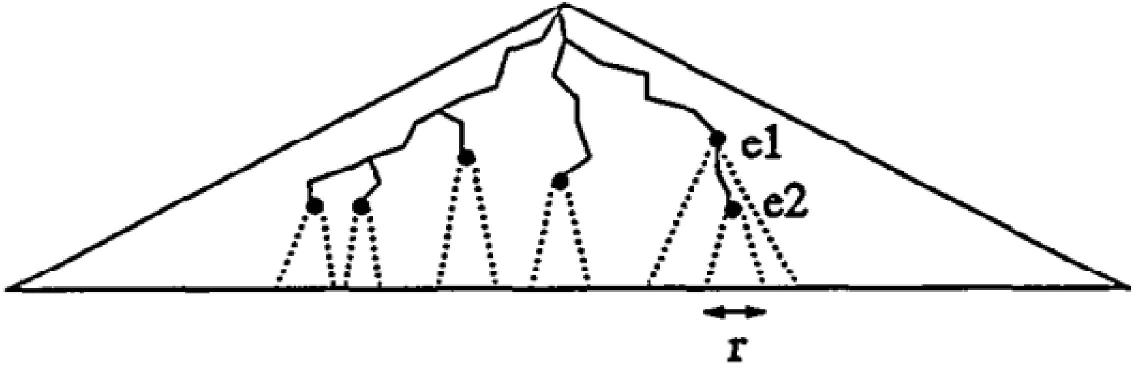


Figure 4: Routing entries defining ranges of IP addresses.

Small forwarding tables for fast routing lookups，SIGCOMM’97

University of Chinese Academy of Sciences

DXR

DXR: Towards a Billion Routing Lookups per Second in Software, SIGCOMM CCR, Oct.2012

	IPv4 prefix	next hop
1:	0.0.0.0/0	A
2:	1.0.0.0/8	B
3:	1.2.0.0/16	C
4:	1.2.3.0/24	D
5:	1.2.4.5/32	C

① 路由转发表

	IPv4 address range	next hop	(prefix)
1:	0.0.0.0 .. 0.255.255.255	A	(1)
2:	1.0.0.0 .. 1.1.255.255	B	(2)
3:	1.2.0.0 .. 1.2.2.255	C	(3)
4:	1.2.3.0 .. 1.2.3.255	D	(4)
5:	1.2.4.0 .. 1.2.4.4	C	(3)
6:	1.2.4.5 .. 1.2.4.5	C	(5)
7:	1.2.4.6 .. 1.2.255.255	C	(3)
8:	1.3.0.0 .. 1.255.255.255	B	(2)
9:	2.0.0.0 .. 255.255.255.255	A	(1)

② 空间映射，构建非重叠区间

	IPv4 address range	next hop	(prefix)
1:	0.0.0.0 .. 0.255.255.255	A	(1)
2:	1.0.0.0 .. 1.1.255.255	B	(2)
3:	1.2.0.0 .. 1.2.2.255	C	(3)
4:	1.2.3.0 .. 1.2.3.255	D	(4)
5:	1.2.4.0 .. 1.2.255.255	C	(3,5)
6:	1.3.0.0 .. 1.255.255.255	B	(2)
7:	2.0.0.0 .. 255.255.255.255	A	(1)

③ 连续相同下一跳规则区间合并

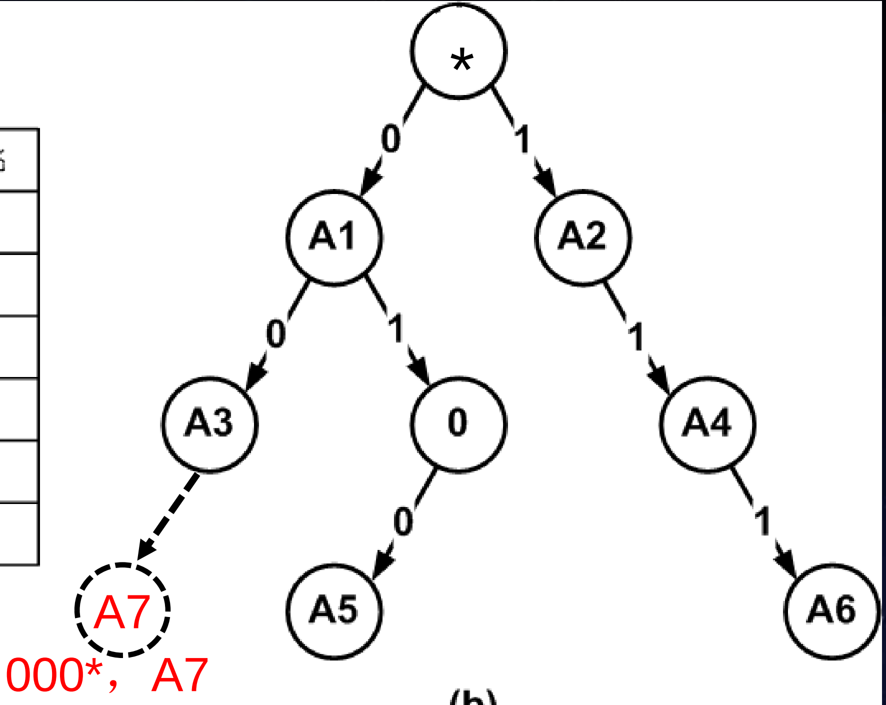
	interval base	next hop
1:	0.0.0.0	A
2:	1.0.0.0	B
3:	1.2.0.0	C
4:	1.2.3.0	D
5:	1.2.4.0	C
6:	1.3.0.0	B
7:	2.0.0.0	A

④ 以起始点代替上一条规则终点

Trie查找

前缀	下一跳
0*	A1
1*	A2
00*	A3
11*	A4
010*	A5
111*	A6

(a)



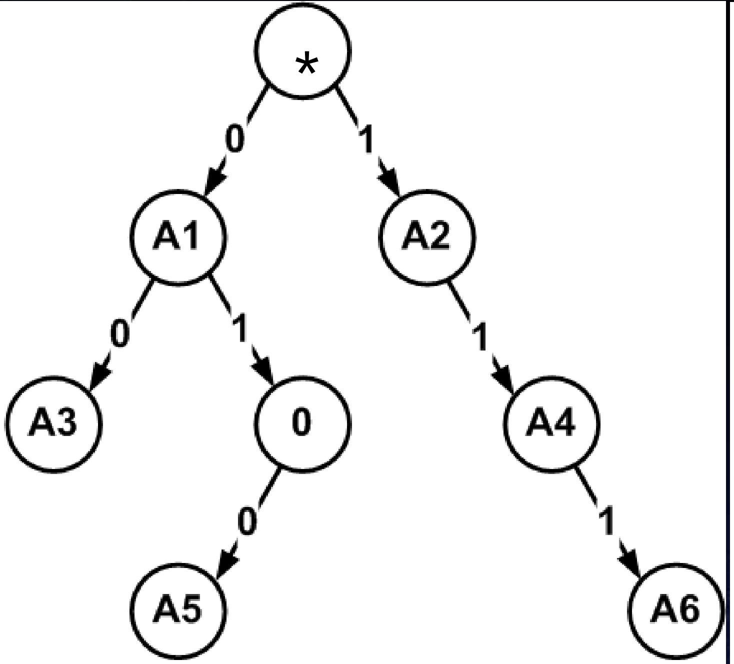
(b)

- Trie构建过程
- LPM/Insert/Del.
- 性能
  - Tire存储空间
  - 查找过程内存访问次数
  - 更新开销

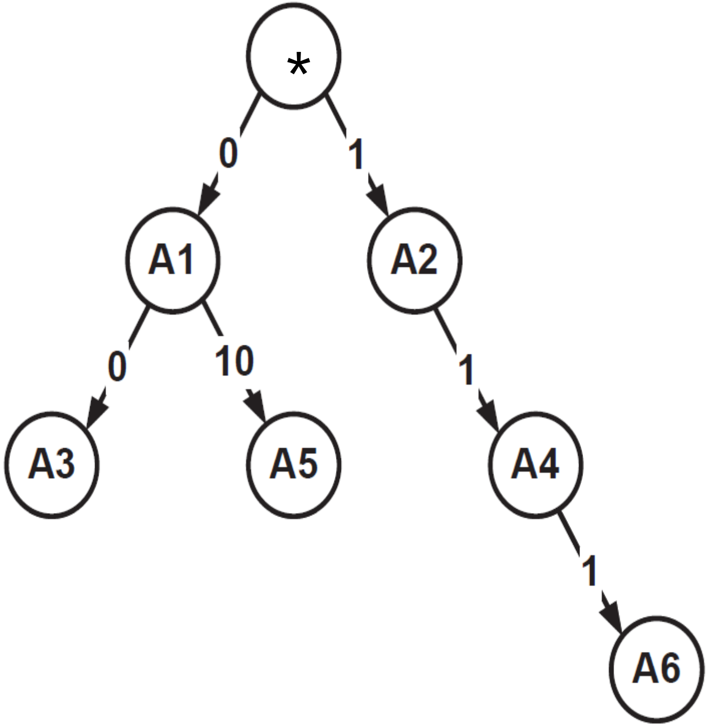
Trie查找改进：路径压缩

前缀	下一跳
0*	A1
1*	A2
00*	A3
11*	A4
010*	A5
111*	A6

(a)



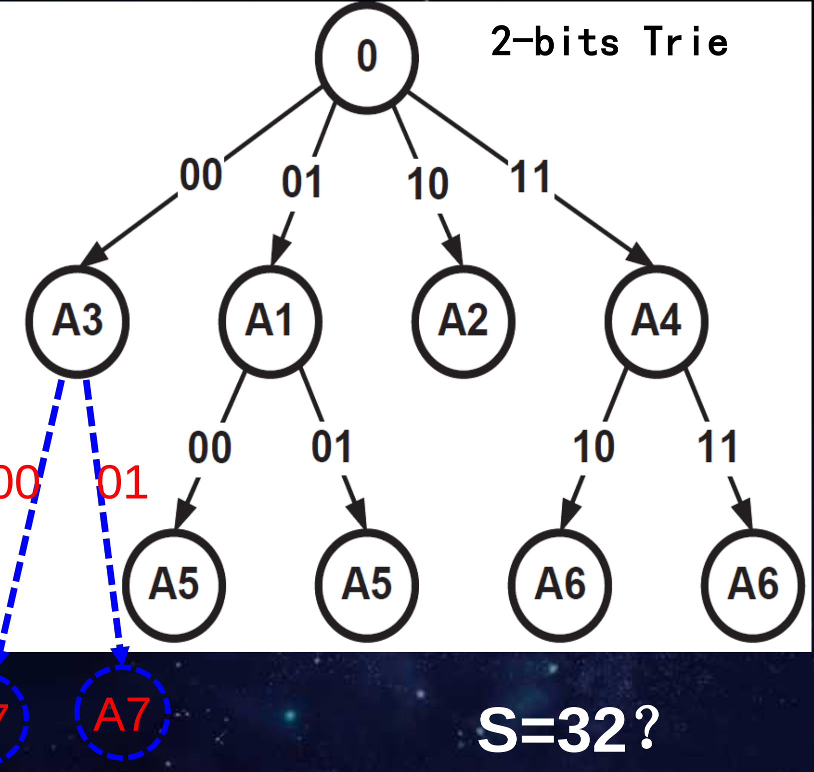
(b)



稀疏Trie：减少存储节点、缩短访问路径；数据结构，更新

Trie查找改进：多比特Trie

前缀	下一跳
0*	A1
1*	A2
00*	A3
11*	A4
010*	A5
111*	A6

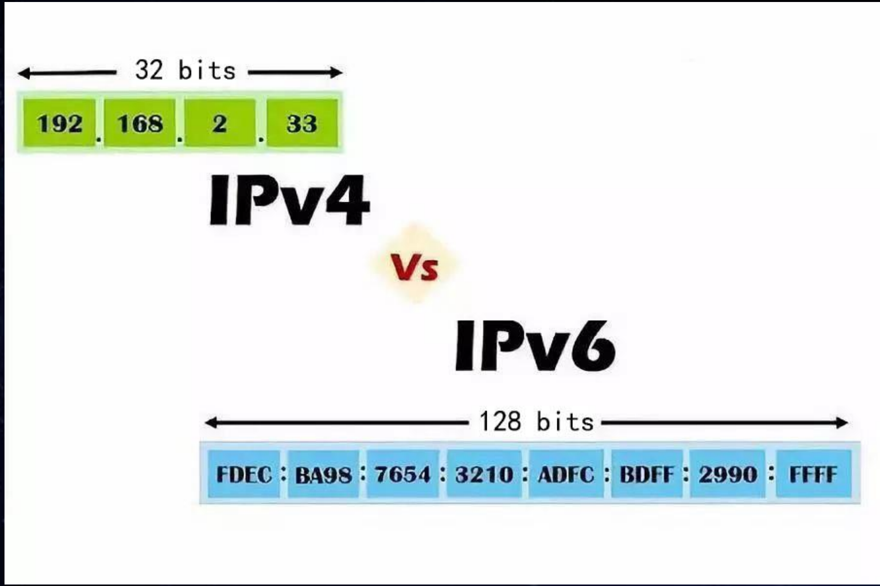


000*，A7

S=32？

- 多比特Trie
  - 查找性能：O(W/S)
  - 存储空间⇧
    - 每个节点：2^S
    - 规则冗余复制
  - 更新 ⇧
    - Insert
    - Del.

IPv6路由查找问题



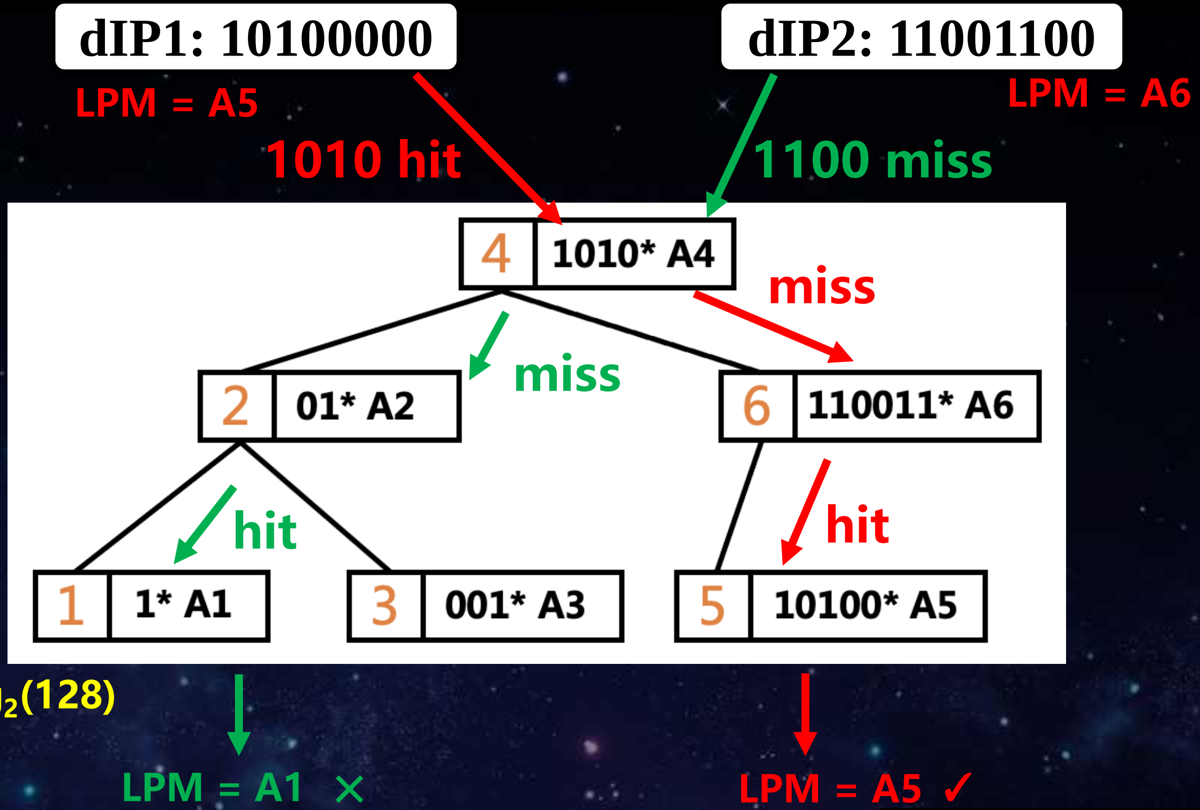
- 海量地址空间
  - $2^{32} \rightarrow 2^{128}$
  - 搜索空间增大
- 更长前缀长度
  - $32 \rightarrow 128$
  - 搜索路径变长

存储开销↑

查找性能↓

IPv6查找：基于前缀长度二分搜索的Hash查找

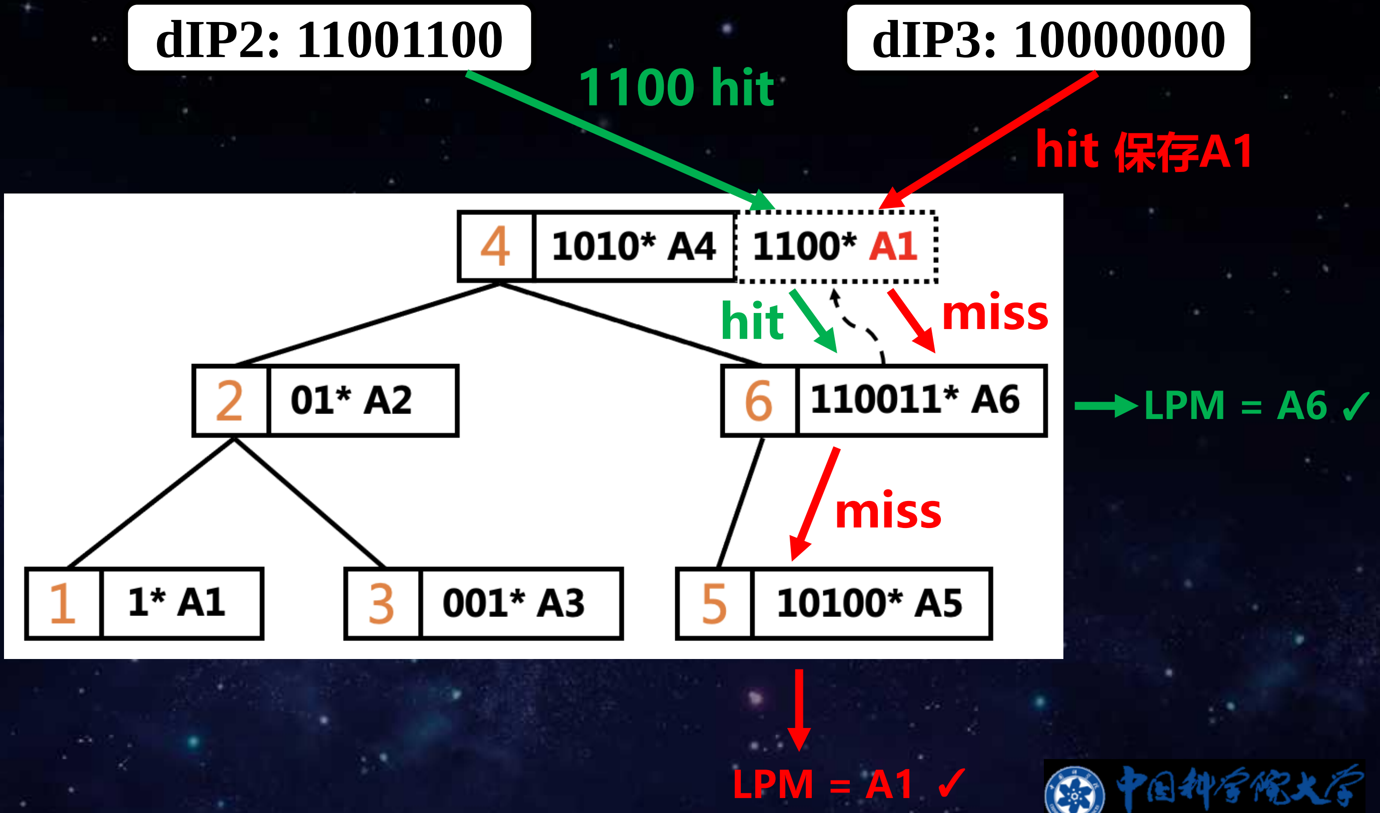
前缀	下一跳
1*	A1
01*	A2
001*	A3
1010*	A4
10100*	A5
110011*	A6



- 按前缀长度构建二分搜索树~$\log_2(128)$
- 命中，右
- 不名中？右孩子节点上移？

IPv6查找：基于前缀长度二分搜索的Hash查找

前缀	下一跳
1*	A1
01*	A2
001*	A3
1010*	A4
10100*	A5
110011*	A6



- 左孩子节点上移
- 更新问题

查找算法性能决定因素：时间与空间复杂度?

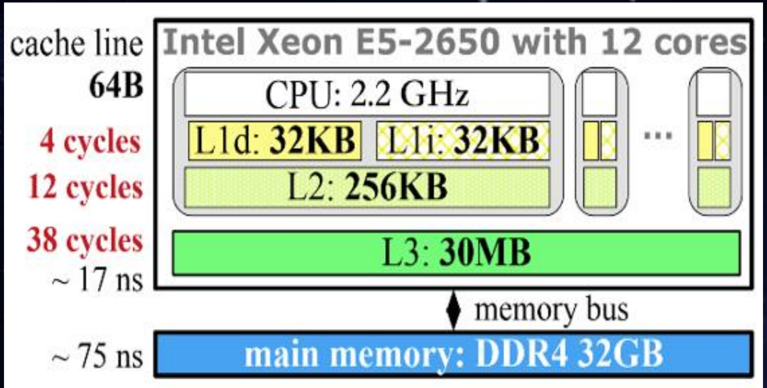
University of Chinese Academy of Sciences

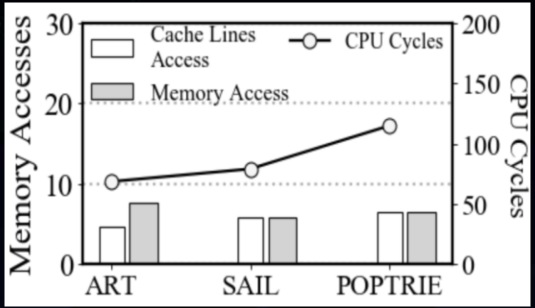
提升查找性能新思路?

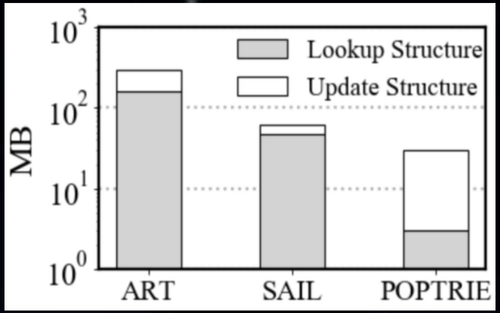
- 当前最快的几个算法及优化思路
  - 减少时间复杂性到极致: Sail (SIGCOMM14)
    - 一次查找只需要1-2次访存和大约20 条CPU 指令
  - 存储开销的极致压缩: DxR (SIGCOMM CCR 12)
    - 压缩整个数据结构，使得其可以塞进 L3
  - 在时空复杂性上进行折衷考虑和优化: poptrie (SIGCOMM 15)
    - 一次查找最坏也只需5-6次访存
    - 大部分结构可以存放在L3

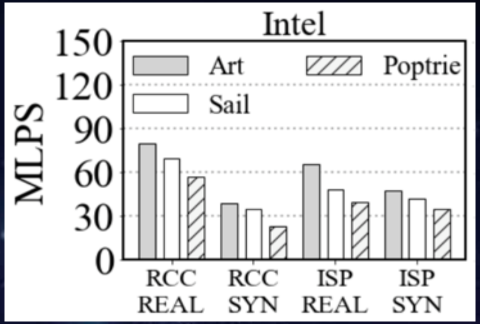
算法设计：硬件特性

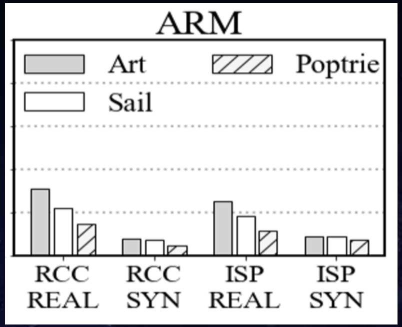
- 时空复杂度：性能决定性因素?





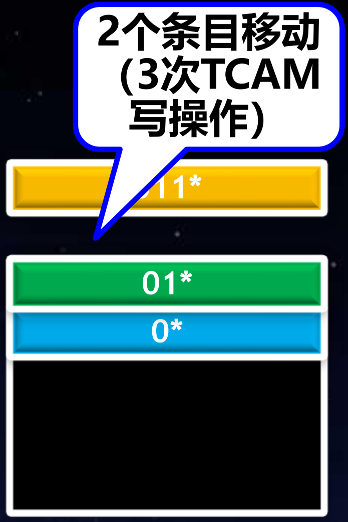




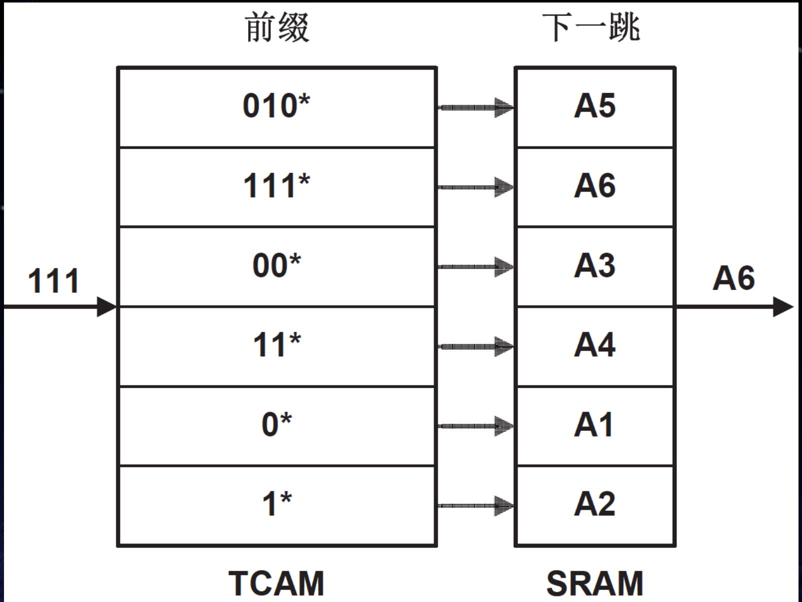


基于硬件的查找方法

TCAM中的前缀顺序限制



插入一条前缀011*

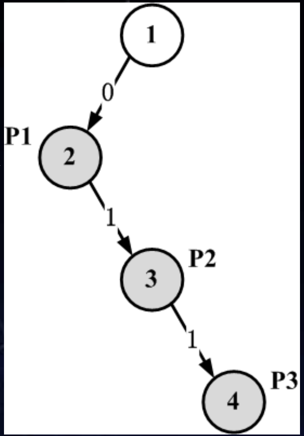


地址空间重叠的前缀需要有顺序关系

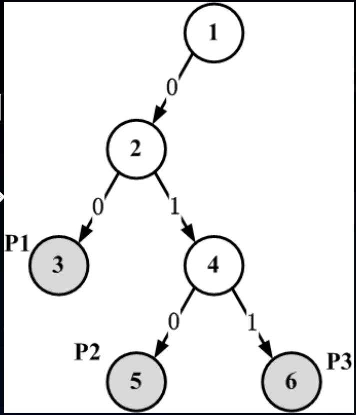
构建不相交的前缀集合：Leaf Pushing

prefix:
P1: 0*
P2: 01*
P3: 011*

Trie built



leaf pushing

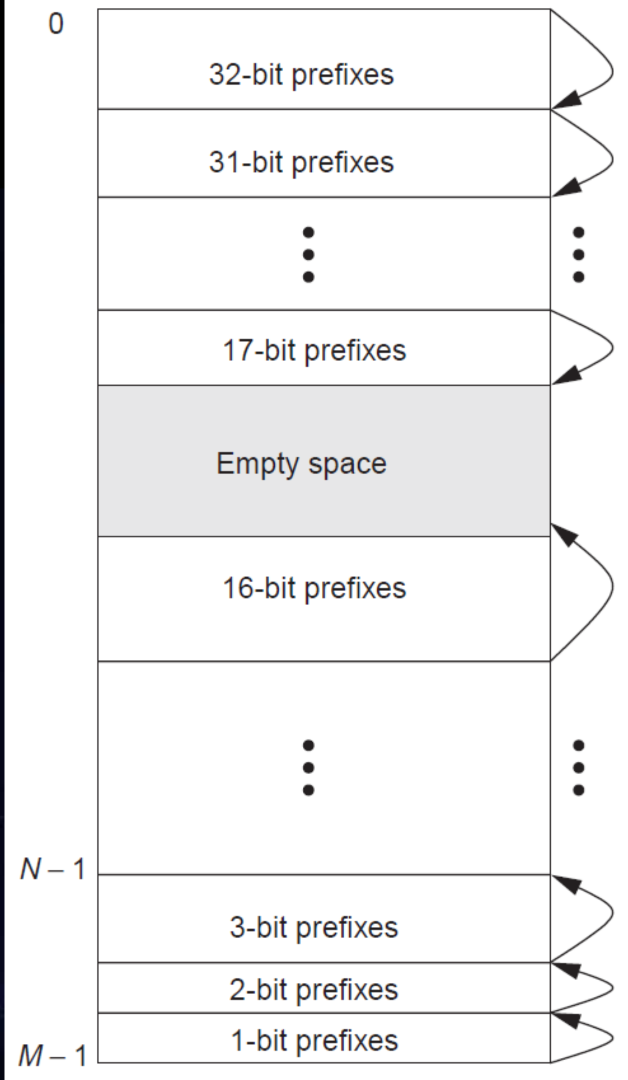


corresponding prefixes

P1: 00*
P2: 010*
P3: 011*

Insert 000*, P4

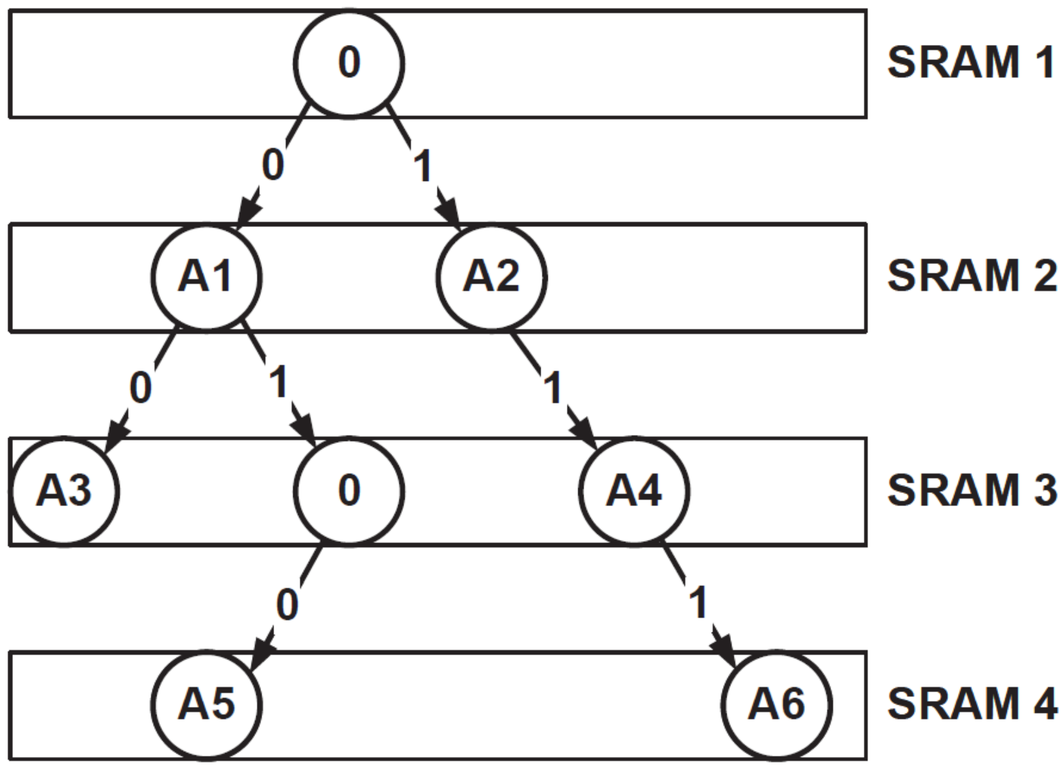
P4：000*
P1：001*
P2：010*
P3：011*



PLO_OPT方法
Insrt：w/2
W：前缀长度

每个长度段之间预留空间?

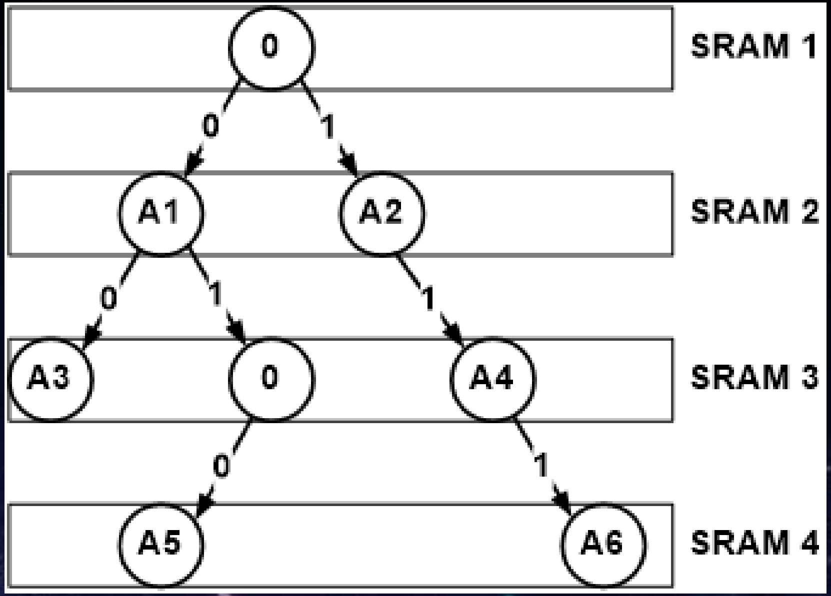
Trie查找：FPGA流水线



- 查找速度⇧
- 更新速度⇧
- 问题
  - 每一级SRAM如何分配
  - FPGA中SRAM容量

硬件查找的优缺点

- TCAM
  - 确定性高速查找，一时钟周期
  - 更新开销大
- FPGA
  - 将trie树每级节点映射到一个流水级
  - 片内SRAM流水线存储空间不足

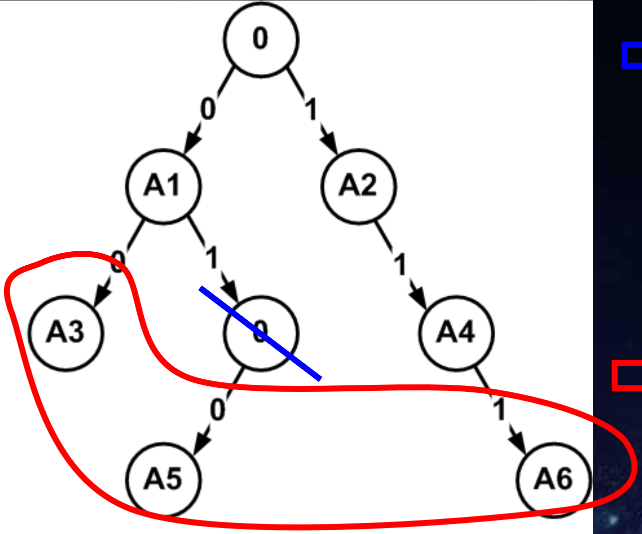


Trie分割

- 叶子前缀（90%）自然不相交[1]
- 裁剪后的小trie树(12%)

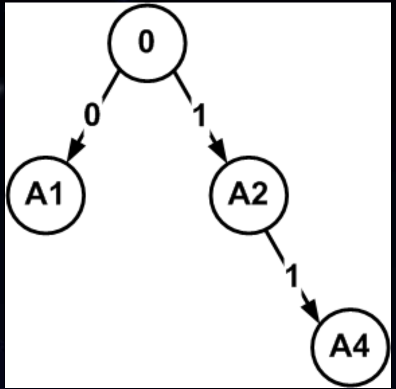
前缀	下一跳
0*	A1
1*	A2
00*	A3
11*	A4
010*	A5
111*	A6

(a)



(b)

(a) FIB　　(b)对应的1-bit trie树



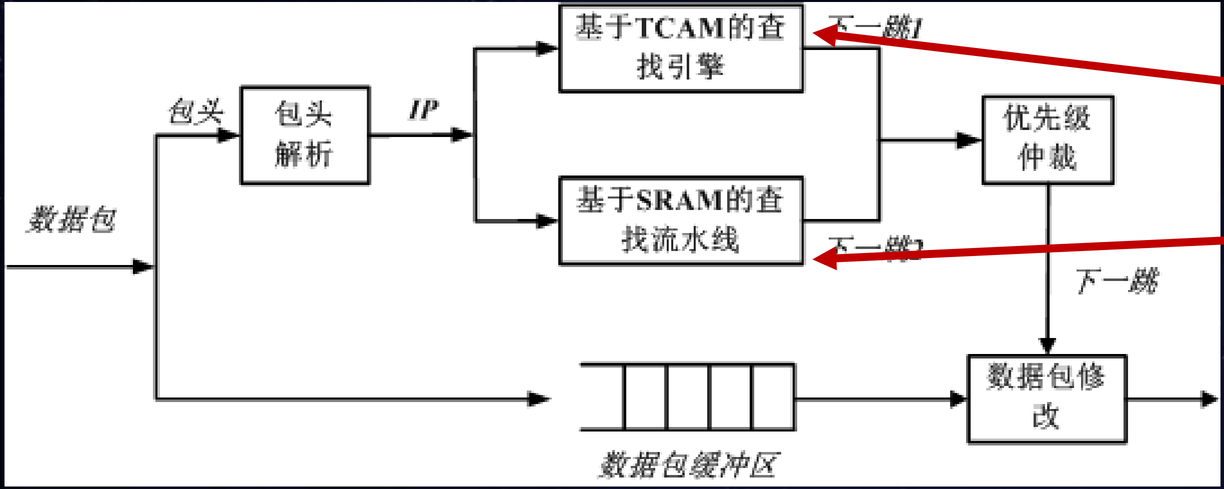
重叠trie树
（重叠前缀集）

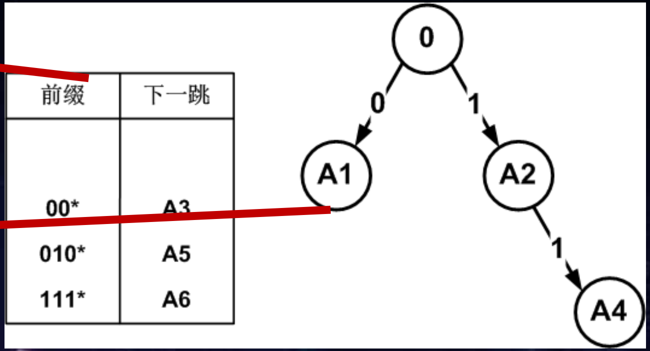
前缀	下一跳
00*	A3
010*	A5
111*	A6

不相交前缀集

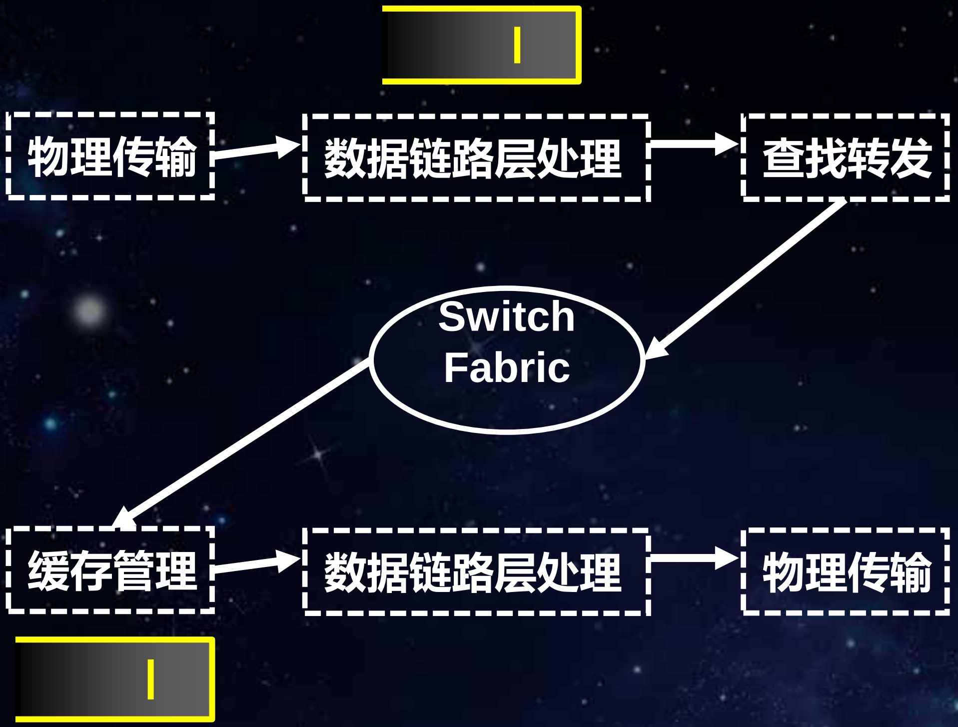
混合的IP查找架构

- 不相交前缀集 -> TCAM（没有前缀顺序限制，更新开销小）
- 重叠小trie树-> FPGA片内SRAM流水线（存储空间需求小）
- 优先级仲裁缓存实现查找引擎同步





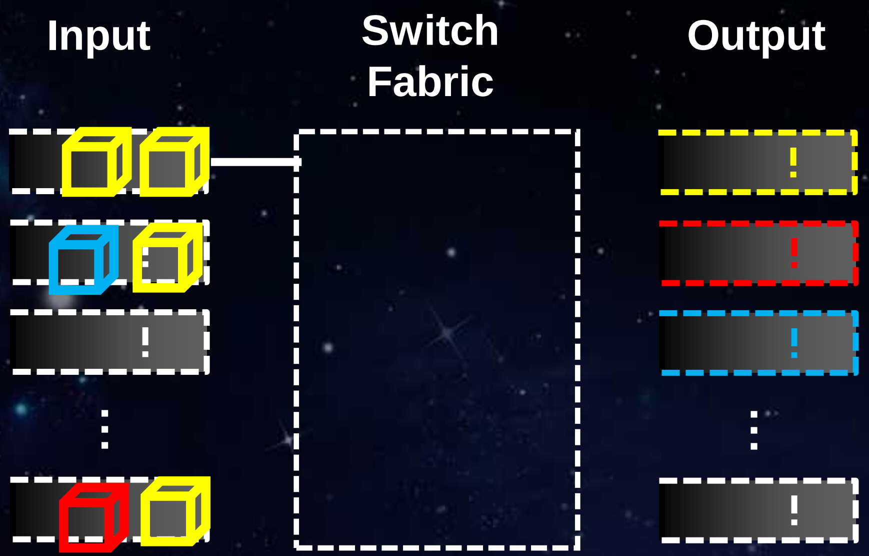
性能模型



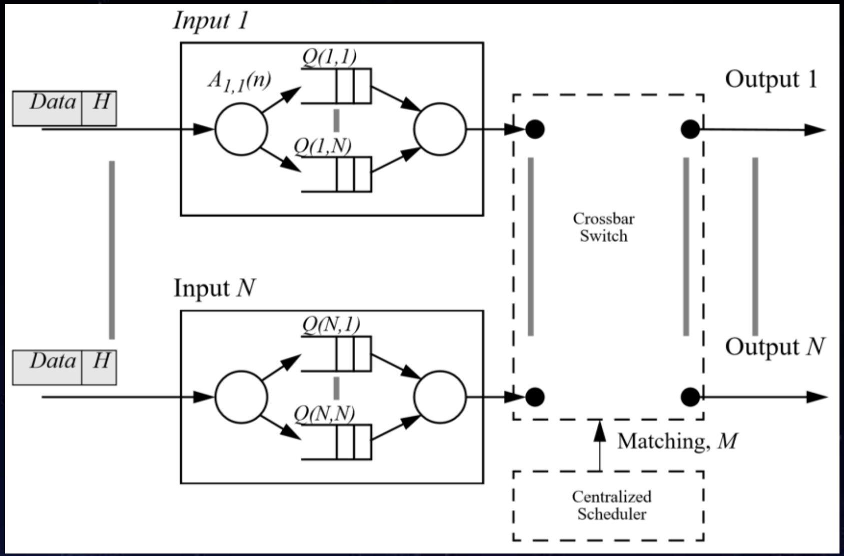
- 排队模型：X/Y/Z
  - FCFS
  - X: 数据包达到间隔时间分布
  - Y：数据包查找转发时间
  - Z：并行查找引擎数量
- 性能模型
  - 数据包Poisson流：$p_n(t)=\frac{(\lambda t)^n}{n!}e^{-\lambda t}$
  - 数据包查找转发时间μ负指数分布
  - 队列长度 $L_s=\frac{\lambda}{\mu-\lambda}$
  - 排队时间 $W_s=\frac{1}{\mu-\lambda}$

中国科学院大学 University of Chinese Academy of Sciences

输入队列



Head-of-the-line Blocking

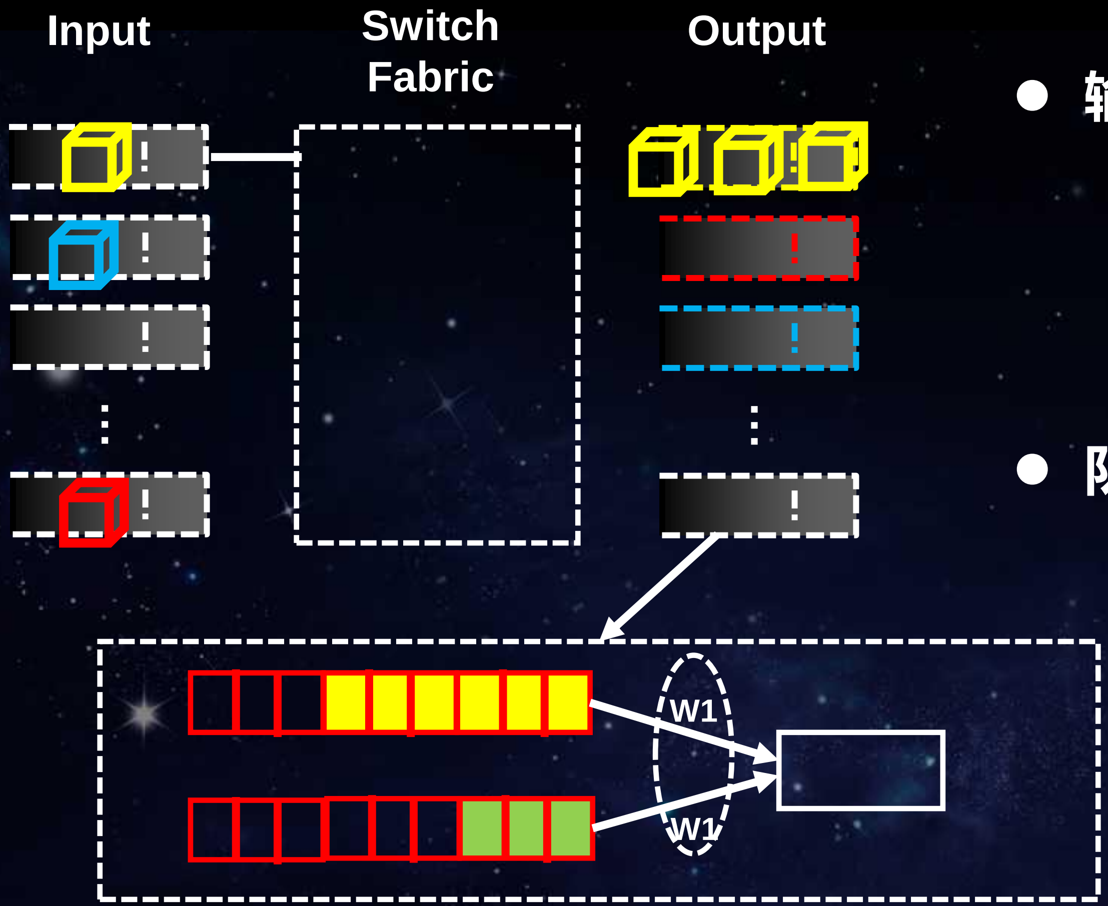


HOL Blocking: VOQ

Virtual Output Queuing

N. McKeown; V. Anantharam; J. Walrand, Achieving 100% Throughput in an input-queued switch,Infocom96

输出队列



- 输出队列缓存溢出，丢包策略
  - Drop-tail
  - Active Queue Management AQM
    - Random Early Detection RED
- 队列管理
  - Round Robin and Weighted Fair Queuing

队列管理策略影响因素：流量模型与QoS

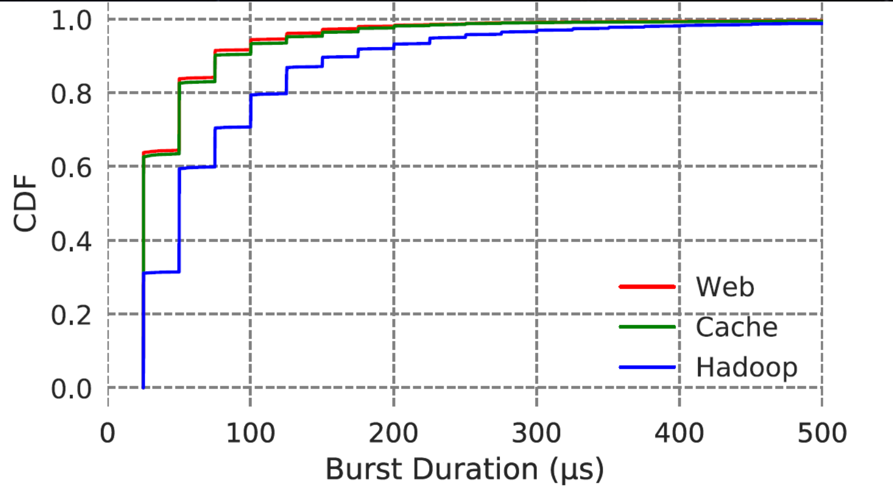
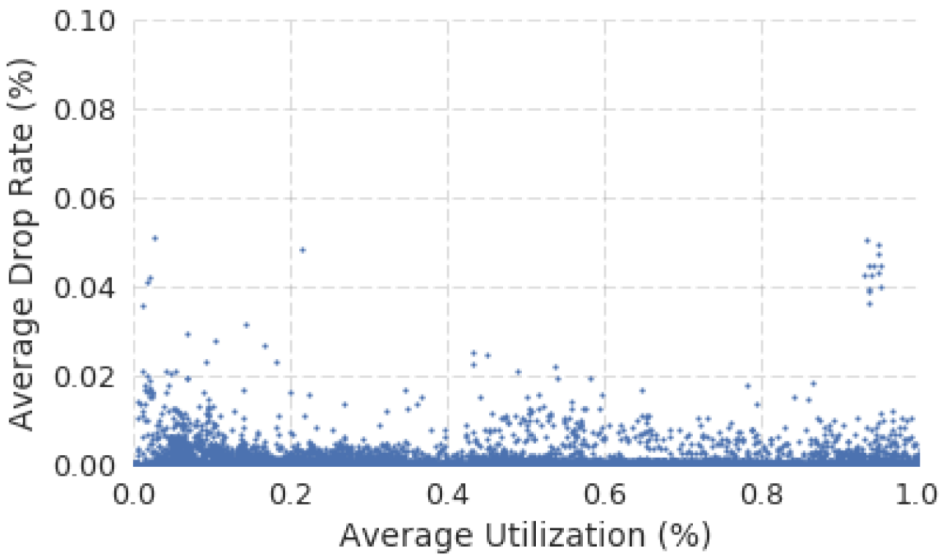
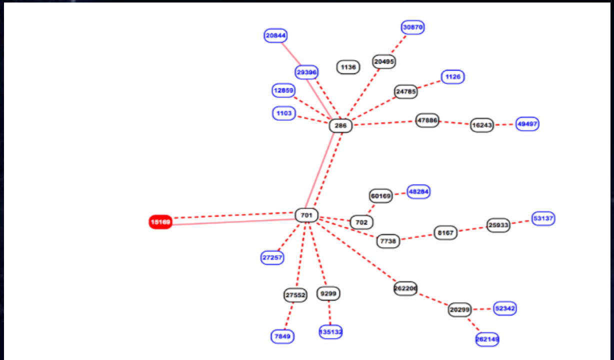


Figure 3: CDF of μburst durations at a 25 μs granularity.



High Resolution Measurement of Data Center Microbursts
IMC 2017，from U. Washington and Facebook

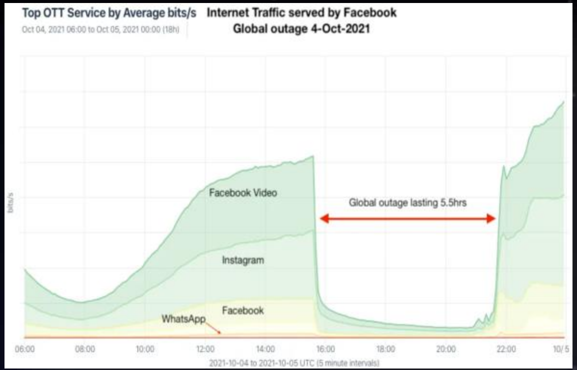
路由安全问题



2017年，Google路由配置错误，造成日本大规模断网1小时

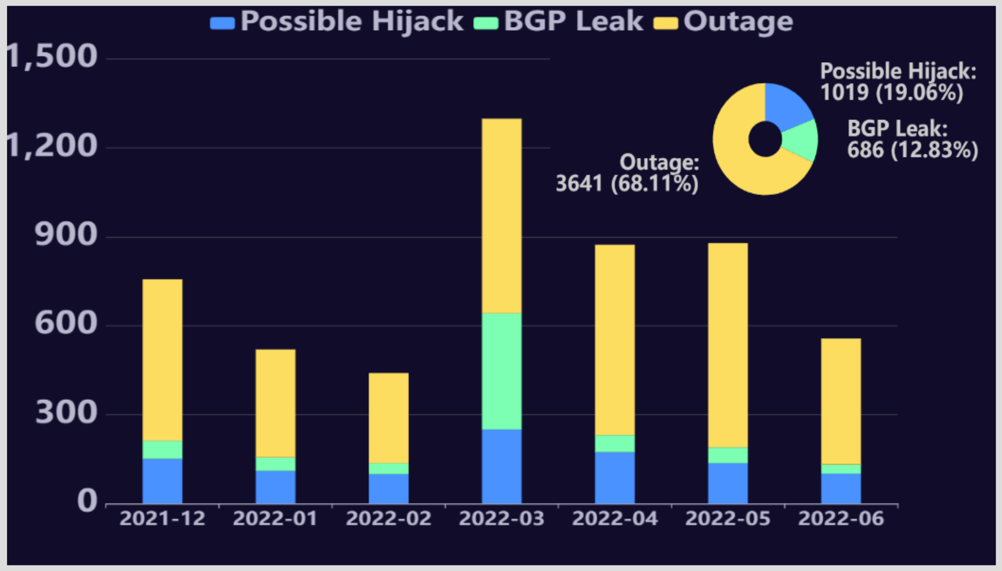
https://www.bgpmon.net/

2018 亚马逊路由被劫持导致DNS被污染，加密货币窃取



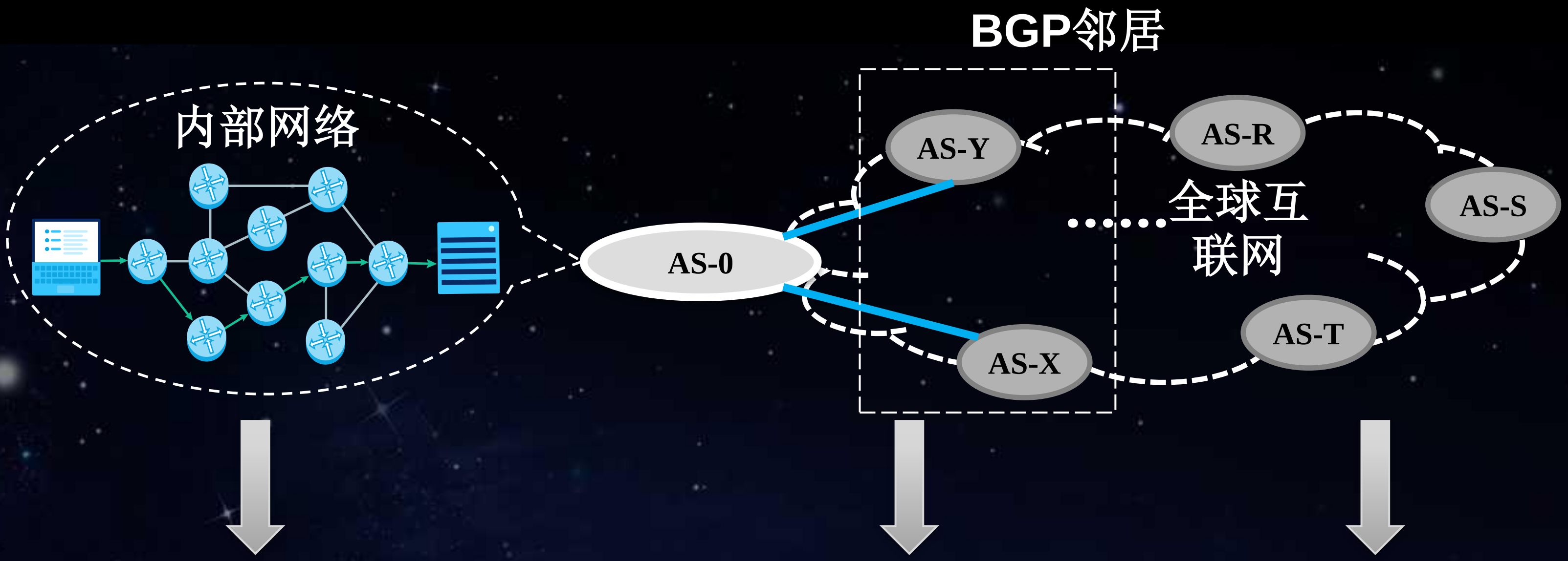
2021年，FB DNS服务器的BGP路由被撤销，导致“史上最严重宕机”

路由安全问题



数据来源： BGPStream, https://www.bgpstream.com/

BGP路由潜在风险



风险一：配置错误

内部网路配置错误或操作失误引发外部BGP路由异常。如2021年Facebook宕机和韩国KT断网等

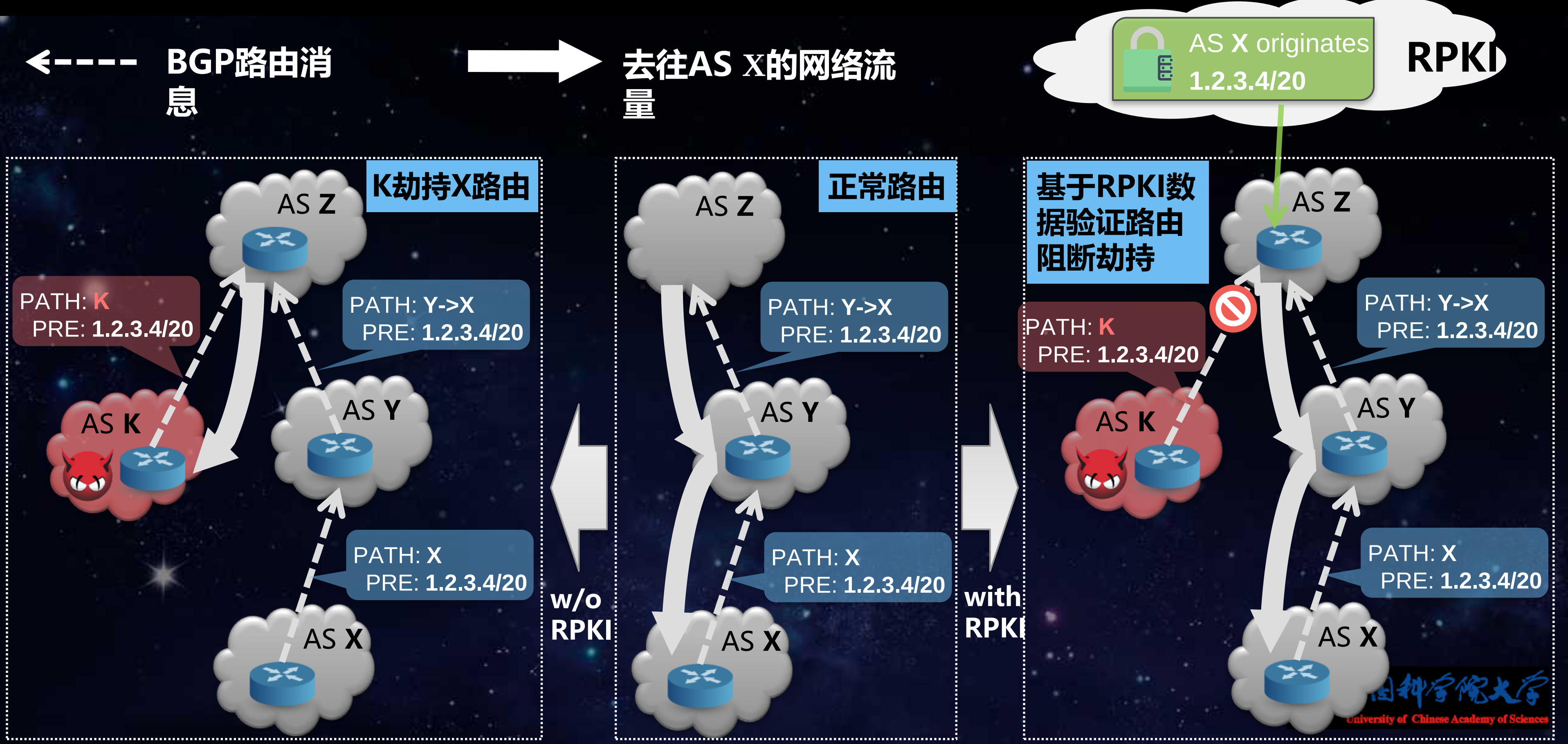
风险二：邻居泄露

邻居发生或传递BGP路由泄露，本网受污染后“劫持”外部流量，挤占带宽资。

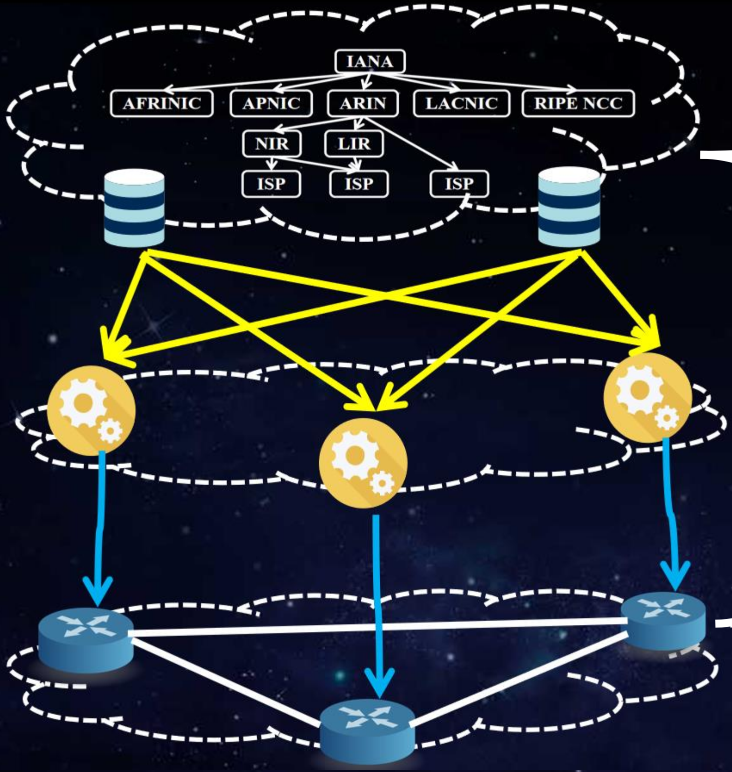
风险三：恶意攻击

外部攻击者通过前缀劫持和路径篡改等方式劫持BGP路由，导致断网或业务被监听破坏。如2018年亚马逊路由被劫持

RPKI：可信数据驱动路由验证



RPKI大规模部署问题



制约大规模部署的关键问题

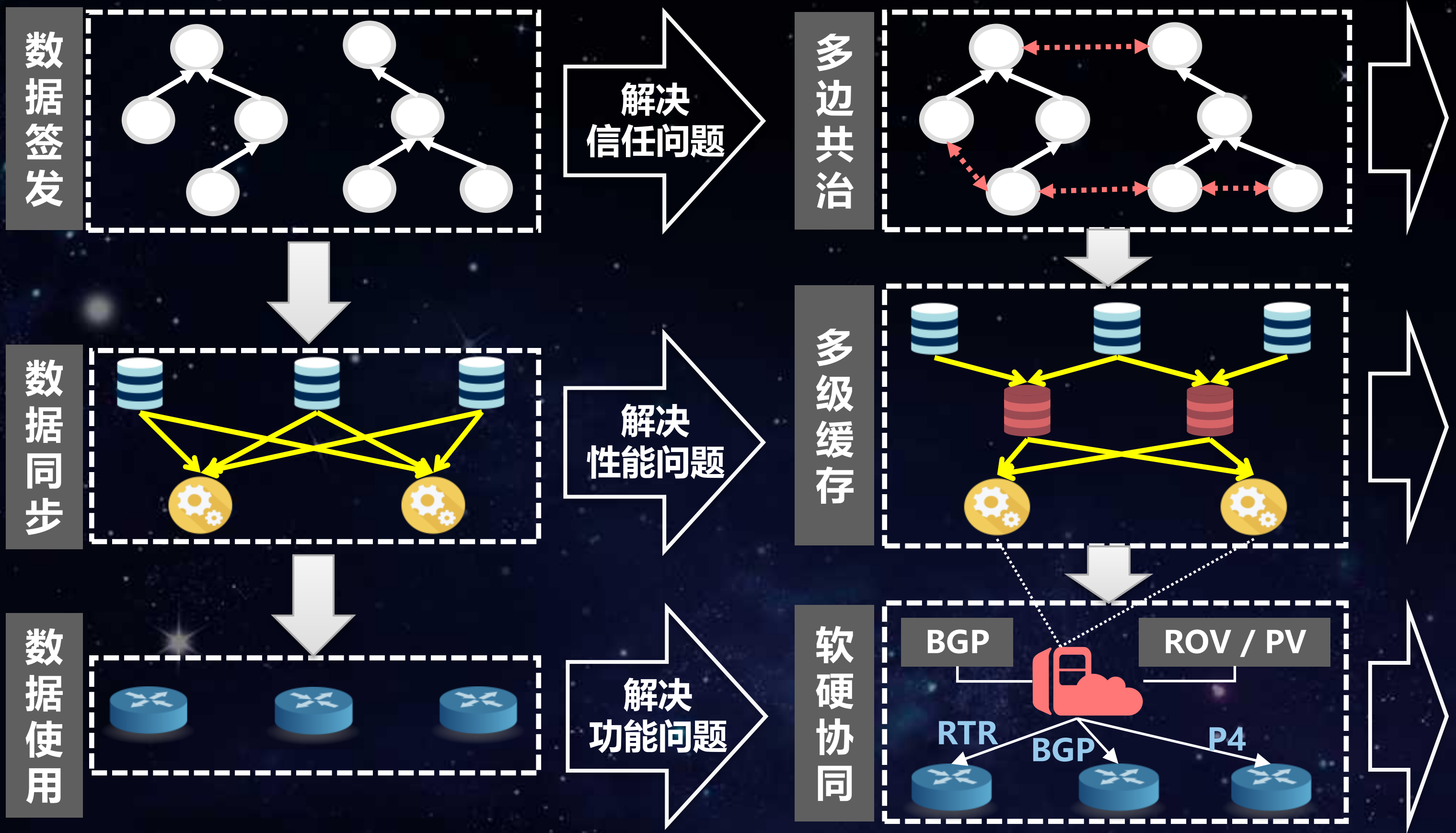
信任
- 树型信任模型：上下级权力严重失衡
- 签发服务托管：降低恶意操作门槛

性能
- 资料库数据需同步到所有RP：~10万+
- RP需全局信息：33个点，最慢 3小时[1]
- 路由器从RP拉数据并使用：通信、维护

[1] rcynic summary, https://www.hactrn.net/opaque/rcynic/index.html

功能
- 路径验证及更多：开销、隐私、扩展
- ROV假阳性问题：合法路由被误过滤

研究内容



研究问题一：在构建信任模型时如何兼顾层级式管理和多边协同共治

研究问题二：在同步分发数据时如何权衡数据完整性和同步分发开销

研究问题三：在路由检测过滤时如何折衷考虑检测效率和结果可靠性

思考:高效路由查找转发算法设计(Bonus)

（TO：李彦彪，lybmath@cnic.cn）

Tong Yang, Gaogang Xie, Yanbiao Li, Qiaobin Fu, Alex Liu, Qi Li, Laurent Mathy, Guarantee IP Lookup Performance with FIB Explosion, ACM Sigcomm, 2014

课后选读

- Nick MeKeown, Internet Routers: Past Present and Future, June 2006, London. http://tiny-tera.stanford.edu/~nickm/talks/BCSv6.0.ppt (必读)
- Routing Lookups in Hardware at Memory Access Speeds, Infocom’98 (选读)
- DXR: Towards a Billion Routing Lookups per Second in Software, SIGCOMM CCR, Oct. 2012 (选读)
- A Hybrid Hardware Architecture for High-speed IP Lookups and Fast Route Updates, IEEE/ACM ToN’14 (选读)
- Guarantee IP Lookup Performance with FIB Explosion, ACM Sigcomm’14 (选读)
- Partial Order Theory for Fast TCAM Updates, IEEE/ACM ToN’18 (选读)

谢谢！