ROUTING AND FORWARDING

Mario Baldi
www.baldi.info

Routing

Determine a path through the network for packets

Forwarding

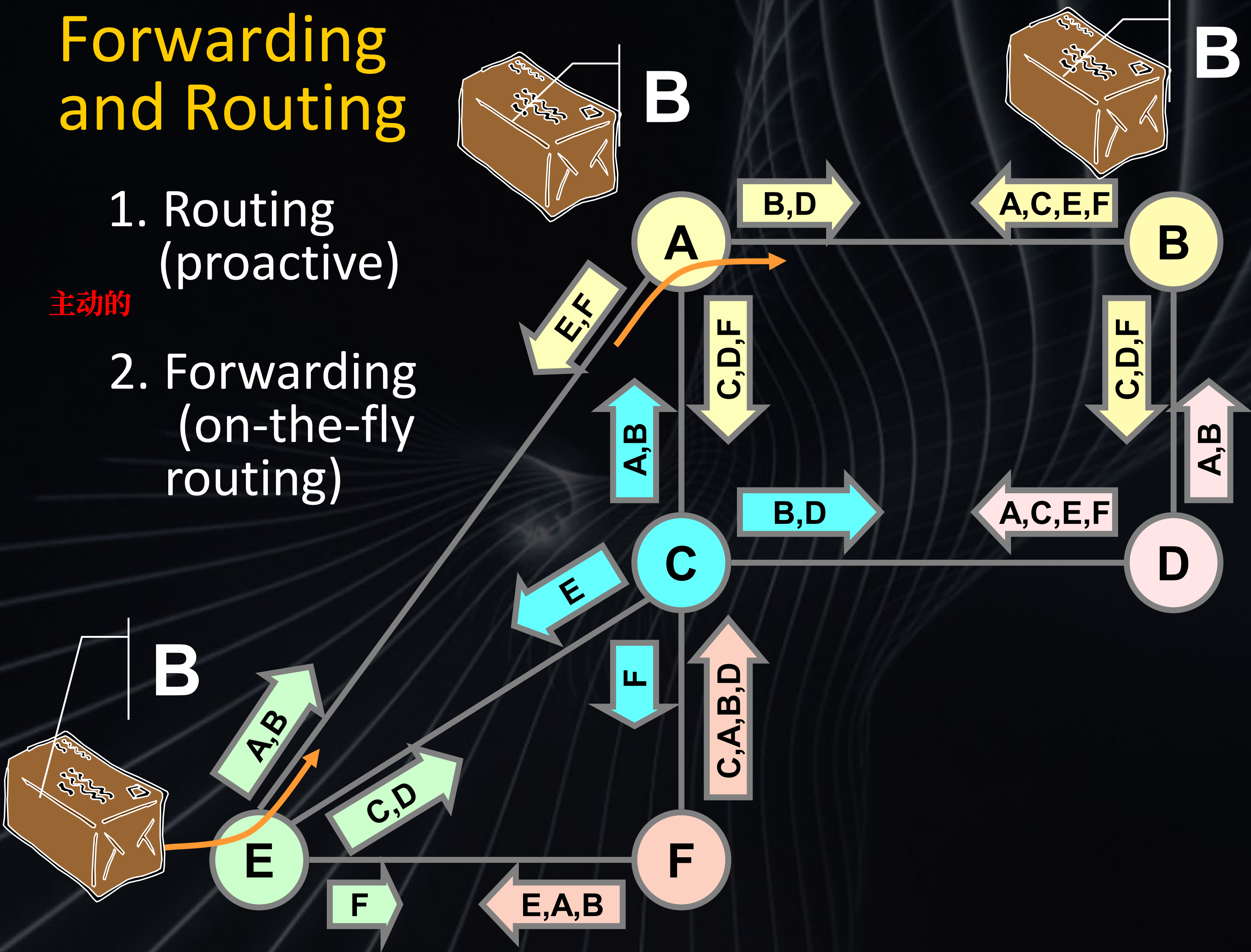
- Advance packets through the network
- Includes a routing decision

Forwarding and Routing

1. Routing (proactive)

主动的

2. Forwarding (on-the-fly routing)



© see page 2

Proactive Routing

- Independent of actual traffic
- Determine reachable destinations
- Compute best route
- Commonly referred to as "routing"

On-the-fly Routing

- Realized when handling each packet
- Based on local information
  - Routing/forwarding table
    - Output of proactive routing or signaling
  - A.k.a. *route*

On-the-fly Routing Algorithms

- → Routing by Network Address
- → Label Swapping
- → Source Routing

Each protocol architecture adopts one or more

Forwarding Phases

- Routing (on-the-fly)
  - Output port selection
  - Possibly next-hop selection
- Switching: transfer to output port
- Transmission

A Proactive Routing Algorithm Classification

主动路由算法分类

- Non-adaptive algorithms (static)
- Adaptive algorithms (dynamic)

NON-ADAPTIVE ROUTING

Non-adaptive Algorithms

不自适应算法

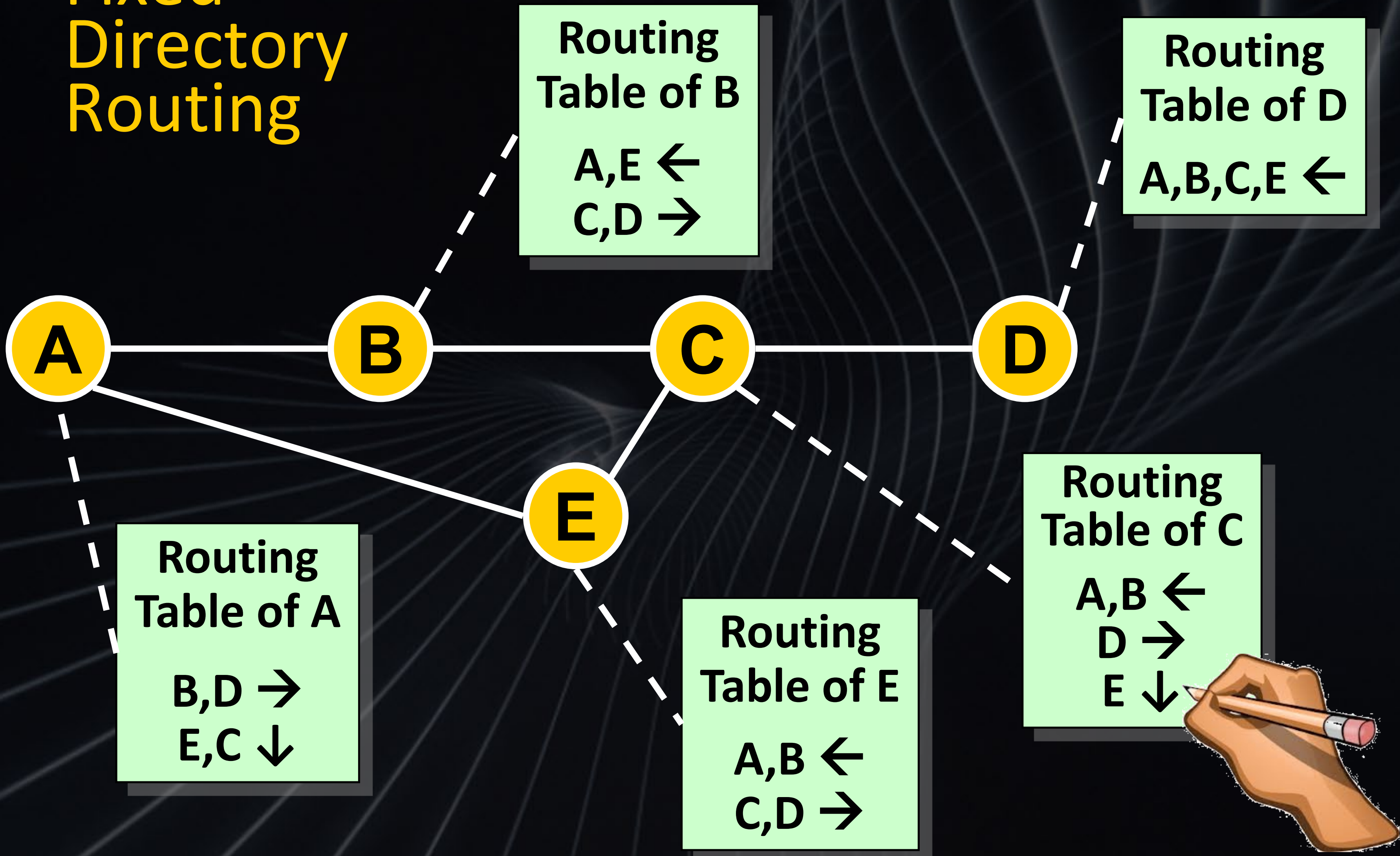
- Fixed Directory routing
  - AKA static routing

    AKA 认证和密钥协商协议
  - Manual configuration

    手动配置
- (Selective) flooding and derivates

固定字典路由

Fixed Directory Routing



优点 和缺点

Pros and Cons

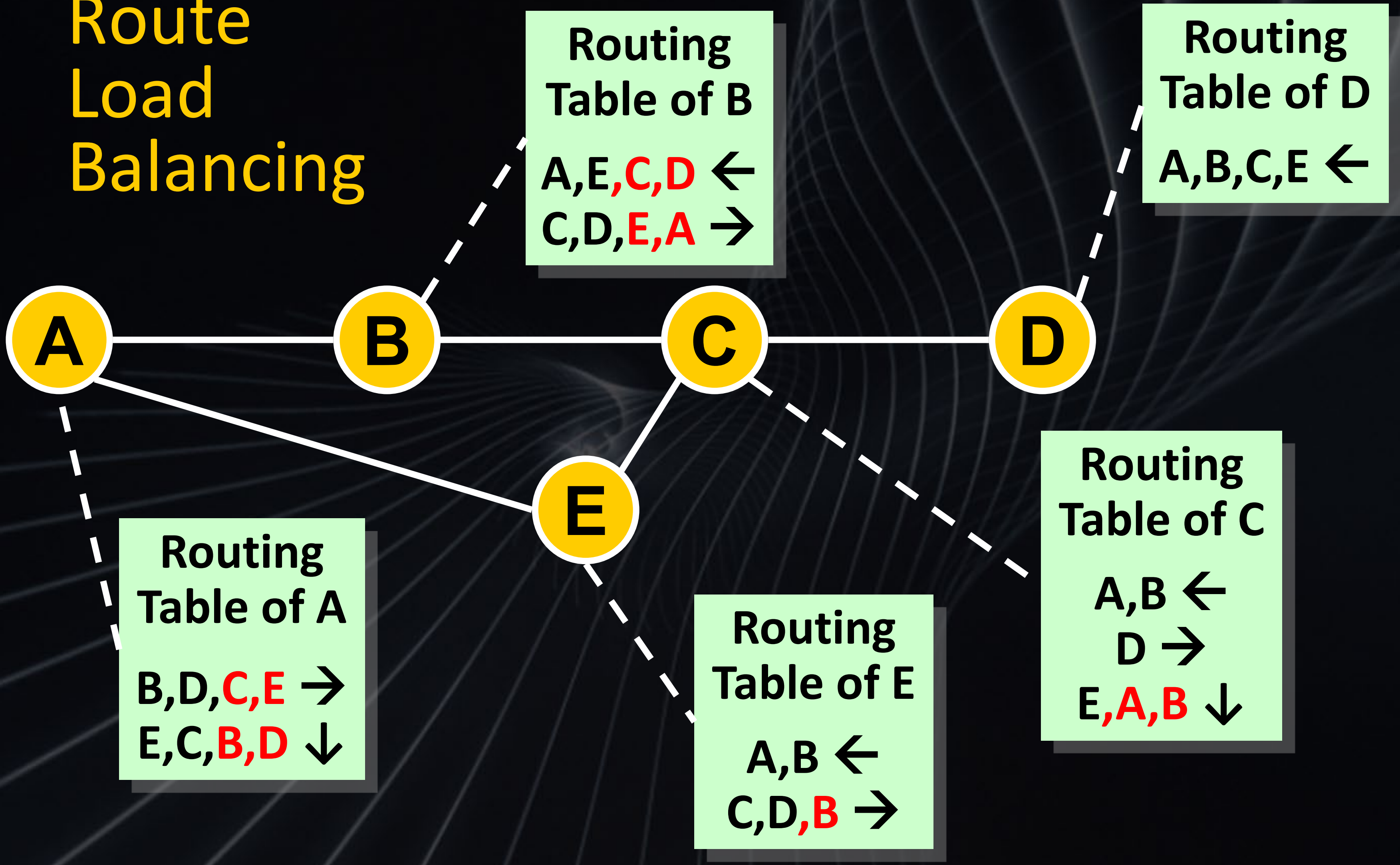
- Administrator has full control 管理员完全控制
- Error prone 易发
- It does not adapt to topology changes

拓扑结构

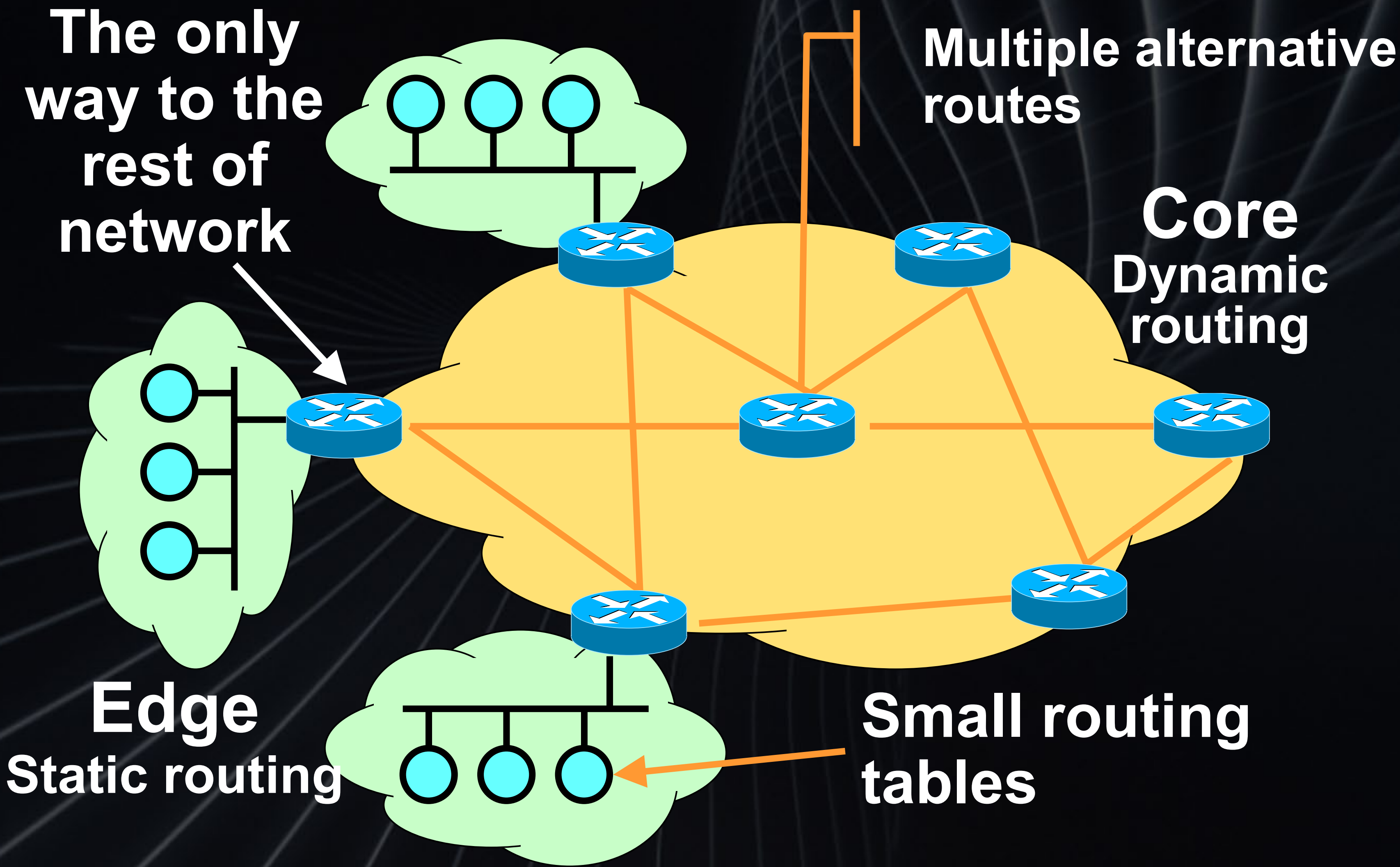
Backup Route Load Balancing

改变拓扑的解决方法

负载均衡



Static vs. Dynamic



DYNAMIC ROUTING

适应算法

Adaptive Algorithms

中心化路由

- Centralized Routing
- Isolated Routing 孤岛路由
- Distributed Routing

分布路由

  - Distance Vector

距离向量

  - Link State

连接状态

Centralized Routing

- Routing Control Center (RCC)
- Calculates and distributes routing tables

计算和分发路由表

- Needs information from all nodes

需要所有结点的信息

Centralized Routing

- Optimizes performance 优化了性能
- Simplifies troubleshooting 简化了排错
- Significant network load in proximity of RCC
  RCC附近负载显著

Centralized Routing

单点故障

- RCC is single point of failure
- RCC is bottleneck 有瓶颈
- Not suitable for highly dynamic networks 大型动态网络

Isolated Routing

节点独自决策

- Each node decides independently
- No exchange of information

不交换数据

- E.g., Backward Learning

后向学习

  - IEEE 802.1D bridges

分布式路由

Distributed Routing

Combines advantages of isolated and centralized routing

结合孤立的和中心化的路由的优点

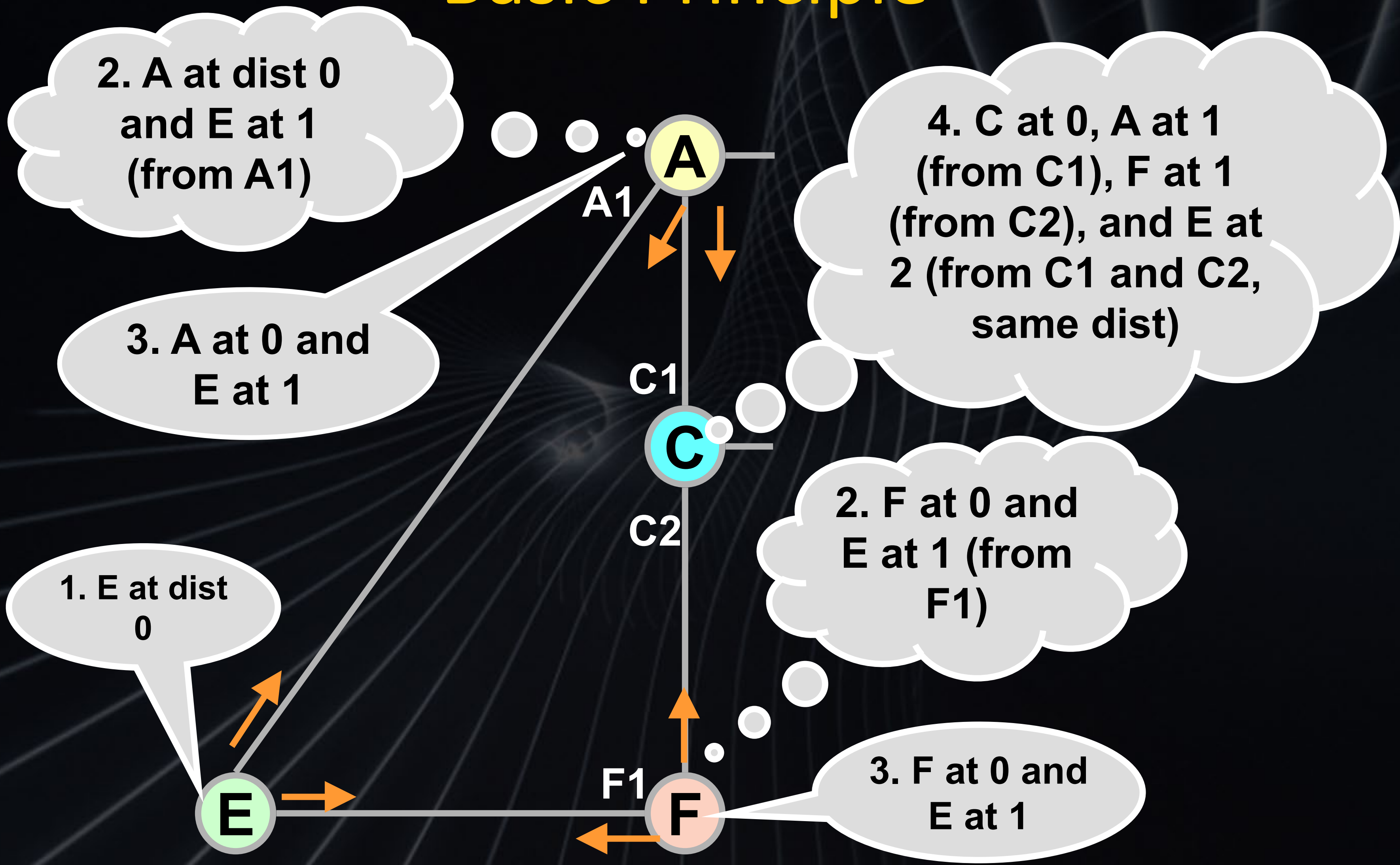
- Routers co-operate in exchanging connectivity information 路由器间互相合作，相互交换了解数据
- Each router decides independently, but coherently

每个路由独自决定， 但是交换信息， 连贯

DISTANCE VECTOR ROUTING ALGORITHM

向量路由算法

Basic Principle



Distance Vector

可到达的地方

- List of reachable destinations (all!)

距离布告路由的距离

- Distance from announcing router
- Generated by each router

由每个路由器生成

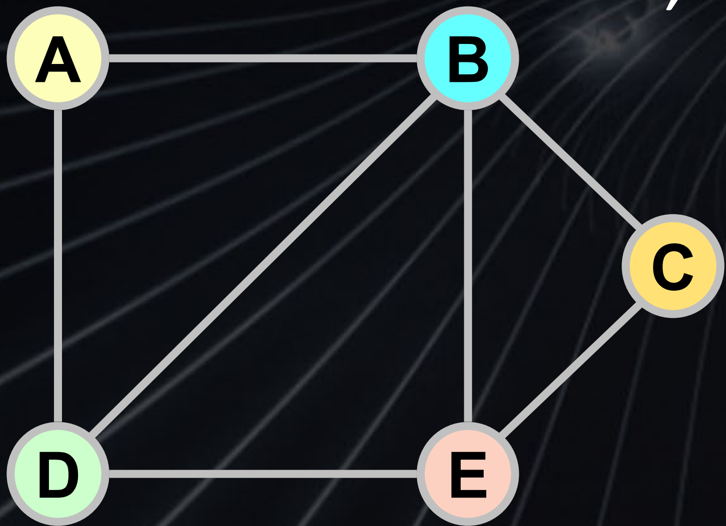
- Received from neighbors

向量从邻居接收

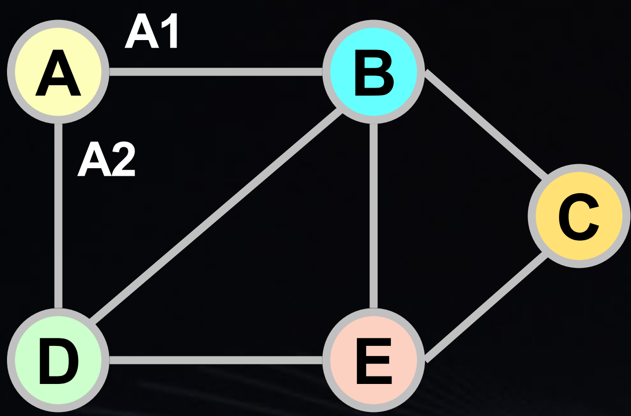
Sample Scenario

情景

Routing information stored by A



Loc (A)	DV (B)	DV (D)
A, 0	A, 1	A, 1
	B, 0	B, 1
	C, 1	C, 2
	D, 1	D, 0
	E, 1	E, 1



Distance Vector Merging and Generation

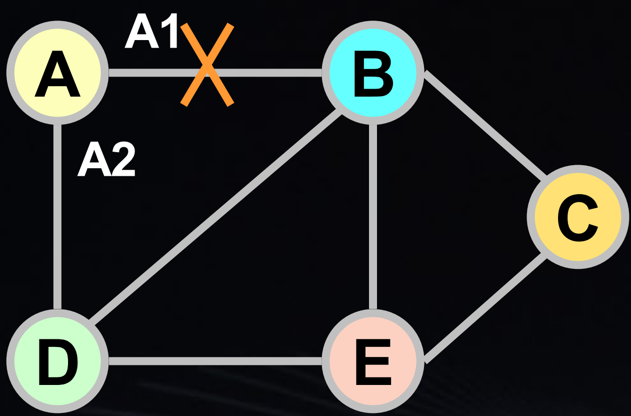
距离向量汇合以及生产

Received from line A1

Received from line A2

Loc (A)	DV (B)	DV (D)	ROUT. TABLE (A)	DV (A)
A, 0	~~A, 1~~	~~A, 1~~	A, local, 0	A, 0
	B, 0	~~B, 1~~	B, A1, 1	B, 1
	C, 1	~~C, 2~~	C, A1, 2	C, 2
	~~D, 1~~	D, 0	D, A2, 1	D, 1
	E, 1	E, 1	E, A2, 2	E, 2

合并的过程中
在以前的
基础上加一



Topology Change

拓扑结构改变

Loc (A)
A, 0

Void!!

DV (B)
~~A, 1~~
~~B, 0~~
~~C, 1~~
~~D, 1~~
~~E, 1~~

DV (D)
~~A, 1~~
B, 1
C, 2
D, 0
E, 1

ROUT. TABLE (A)		
A,	local,	0
B,	A2,	2
C,	A2,	3
D,	A2,	1
E,	A2,	2

DV (A)
A, 0
B, 2
C, 3
D, 1
E, 2

Example: Cold Start

冷开机

RT (A)	RT (B)	RT (C)	RT (D)	RT (E)
A,loc,0	B,loc,0	C,loc,0	D,loc,0	E,loc,0

A sends its DV

RT (A)	RT (B)	RT (C)	RT (D)	RT (E)
A,loc,0	A,B1, 1	C,loc,0	A,D1, 1	E,loc,0
	B,loc,0		D,loc,0	

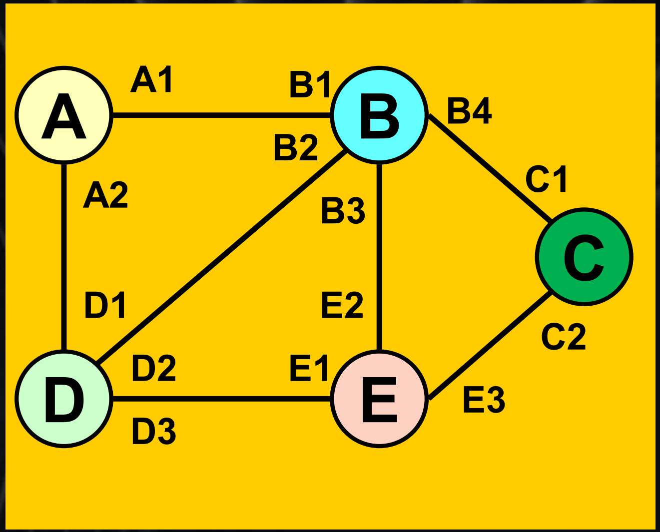
B and D send their DVs

RT (A)	RT (B)	RT (C)	RT (D)	RT (E)
A,loc,0	A,B1, 1	A,C1 ,2	A,D1, 1	A,E2, 2
B,A1, 1	B,loc,0	B,C1 ,1	B,D2, 1	B,E2, 1
D,A2, 1	D,B2, 1	C,loc,0	D,loc,0	D,E1, 1
				E,loc,0

All send their DVs

. . .

RT (A)	RT (B)	RT (C)	RT (D)	RT (E)
A,loc,0	A,B1, 1	A,C1 ,2	A,D1, 1	A,E2, 2
B,A1, 1	B,loc,0	B,C1 ,1	B,D2, 1	B,E2, 1
C,A1, 2	C,B4, 1	C,loc,0	C,D2, 2	C,E3, 1
D,A2, 1	D,B2, 1	D,C2, 2	D,loc,0	D,E1, 1
E,A2, 2	E,B3, 1	E,C2, 1	E,D3, 1	E,loc,0



Issues

- Several problems
  - Black Hole 黑洞，意思是会陷入死循环
  - Count to infinity 计算停不下来
  - Bouncing Effect (loop)

Instability 跳跃效果不稳定

Count to Infinity

IS B:

Loc (B)	~~DV (A)~~	DV (C)		RT (B)
B, 0	~~A, 0~~	A, 2	⇒	A,B2 ,3
	~~B, 1~~	B, 1		B,loc,0
	~~C, 2~~	C, 0		C,B2 ,1

IS C:

Loc (C)	DV (B)	RT (C)
C, 0	A, 1	A,C1 ,2
	B, 0	B,C1 ,1
	C, 1	C,loc,0

B sends DV

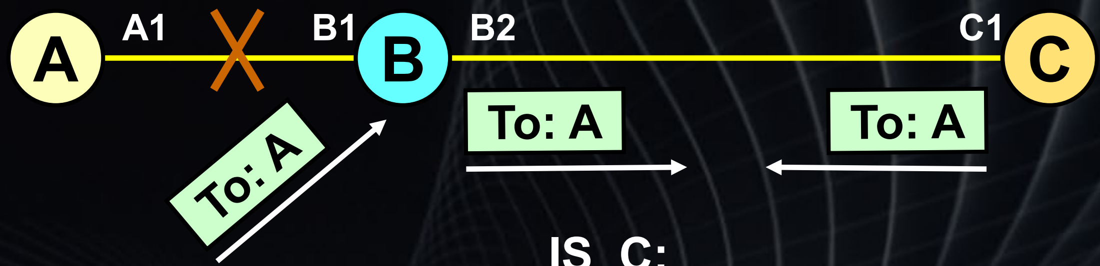
Loc (C)	DV (B)		RT (C)
C, 0	A, 3	⇒	A,C1 ,4
	B, 0		B,C1 ,1
	C, 1		C,loc,0

C sends DV

Loc (B)	DV (C)		RT (B)
B, 0	A, 4	⇒	A,B2 ,5
	B, 1		B,loc,0
	C, 0		C,B2 ,1

Count to Infinity!

Bouncing Effect



IS B:

Loc (B)	~~DV (A)~~	DV (C)	RT (B)
B, 0	~~A, 0~~	A, 2	A,B2 ,3
	~~B, 1~~	B, 1	B,loc,0
	~~C, 2~~	C, 0	C,B2 ,1

IS C:

Loc (C)	DV (B)	RT (C)
C, 0	A, 1	A,C1 ,2
	B, 0	B,C1 ,1
	C, 1	C,loc,0

B sends its DV

Loc (C)	DV (B)	RT (C)
C, 0	A, 3	A,C1 ,4
	B, 0	B,C1 ,1
	C, 1	C,loc,0

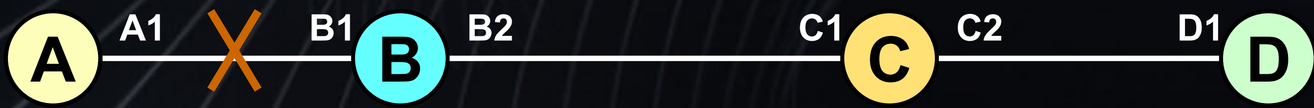
Issues

- Partial solutions 不分解决
  - Split Horizon 水平分裂
  - Path Hold Down
  - Route Poisoning 路由毒化

Split Horizon

"If C reaches destination A through B, it is useless for B trying to reach A through C"

如果c到达目的地A，通过B，那么B尝试通过C到达A是不可行的



DV (C→C1)
C, 0
D, 1

DV (C→C2)
A, 2
B, 1
C, 0

IS B:

Loc (B)	~~DV (A)~~	DV (C)	RT (B)
B, 0	~~A, 0~~	C, 0	B,loc,0
		D, 1	C,B2 ,1
			D,B2, 2

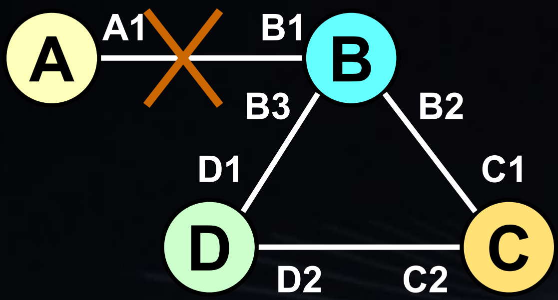
Split Horizon

防止两节点间的循环

- Prevents loops between two nodes
- Speeds up convergence 加速了收敛
- "Personalized" DVs to neighbors
  - DV of C to B does not contain destinations reached through B
- In actual implementations, route has to expire

在实际的应用中，　路由必须有期限

Split Horizon on Mesh



IS C:

Loc (C)	DV (B)	DV (D)	RT (C)
C, 0	A, 1	A, 2	A,C1, 2
	B, 0	B, 1	B,C1, 1
	D, 1	D, 0	C,loc,0
			D,C2, 1

IS D:

Loc (D)	DV (B)	DV (C)	RT (D)
D, 0	A, 1	A, 2	A,D1, 2
	B, 0	B, 1	B,D1, 1
	C, 1	C, 0	C,D2, 1
			D,loc,0

IS B:

Loc (B)	~~DV (A)~~	DV (C)	DV (D)	RT (B)
B, 0	~~A, 0~~	C, 0	C, 1	B,loc,0
		D, 1	D, 0	C,B2 ,1
				D,B3, 1

B sends its DV

IS C:

Loc (C)	DV (B)	DV (D)	RT (C)
C, 0	B, 0	A, 2	A,C2, 3
	D, 1	B, 1	B,C1, 1
		D, 0	C,loc,0
			D,C2, 1

IS D:

Loc (D)	DV (B)	DV (C)	RT (D)
D, 0	B, 0	A, 2	A,D2, 3
	C, 1	B, 1	B,D1, 1
		C, 0	C,D2, 1
			D,loc,0

Split Horizon on Mesh

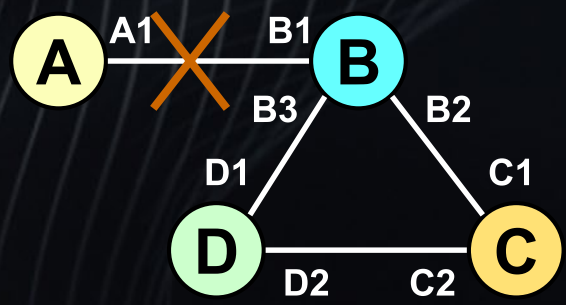
IS C: (from the previous slide)

Loc (C)	DV (B)	DV (D)	RT (C)
C, 0	B, 0	A, 2	A,C2, 3
	D, 1	B, 1	B,C1, 1
		D, 0	C,loc,0
			D,C2, 1

IS D:

Loc (D)	DV (B)	DV (C)	RT (D)
D, 0	B, 0	A, 2	A,D2, 3
	C, 1	B, 1	B,D1, 1
		C, 0	C,D2, 1
			D,loc,0

C and D send their DVs



IS B:

Loc (B)	DV (C)	DV (D)	RT (B)
B, 0	A, 3	A, 3	A,B3, 4
	C, 0	C, 1	B,loc,0
	D, 1	D, 0	C,B2 ,1
			D,B3, 1

IS C:

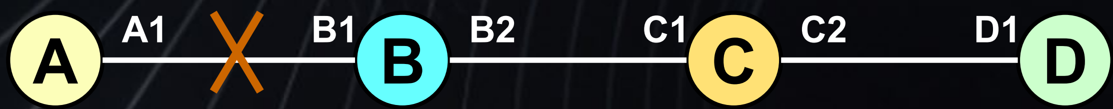
Loc (C)	DV (B)	DV (D)	RT (C)
C, 0	B, 0	B, 1	B,C1, 1
	D, 1	D, 0	C,loc,0
			D,C2, 1

IS D:

Loc (D)	DV (B)	DV (C)	RT (D)
D, 0	B, 0	B, 1	B,D1, 1
	C, 1	C, 0	C,D2, 1
			D,loc,0

Path Hold Down

If link L fails, all destinations reachable through link L are considered unreachable for a certain period of time
I.e., no routes to them are computed



IS B: **Quarantine!**

Loc (B)	~~DV (A)~~	DV (C)		RT (B)
B, 0	~~A, 0~~	~~A, 2~~	⇒	B,loc,0
	~~B, 1~~	B, 1		C,B2 ,1
	~~C, 2~~	C, 0		D,B2, 2
	~~D, 3~~	D, 1		

IS C: **Count to Infinity!**

Loc (C)	DV (B)	DV (D)		RT (C)
C, 0	B, 0	A, 3	⇒	A,C2, 4
	C, 1	B, 2		B,C1, 1
	D, 2	C, 1		C,loc,0
		D, 0		D,C2, 1

Path Hold Down

- High convergence time for the examined node (even with an alternative path)
- The router that noted the fault may not participate to any loop at least until the timeout of Hold Down timer

注意到错误的路由器可能不会参与任何循环，至少直到暂停计时器超时

React on Cost Increase

Routing Poisoning

Routing loops happen when routes have increasing costs

- → Cost-increasing routes in DVs are not used
  - → Two subsequent advertisements show a cost increase
- → Possibly with Path Hold Down 可能会和Path Hold DOwn 一起使用
- → Might block routes with legitimate cost increase

可能会以合理的成本封锁线路

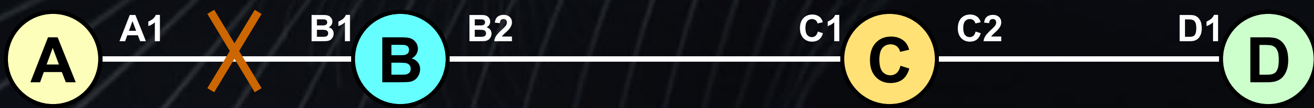
Route Poisoning

An invalid route is advertised at infinite distance/cost 无效的路线以无限距离/无限代价进行广告

- Instead of just omitting it
  - It would have to expire
  - Faster convergence time
- E.g., when link fails or cost increases
- It can substitute or complement Path Hold Down

Split Horizon with Poisonous Reverse

- More aggressive
- No theoretical advantages
- Practically, no need to wait for route expiration



DV (C→C1)
A, inf
B, inf
C, 0
D, 1

DV (C→C2)
A, 2
B, 1
C, 0
D, inf

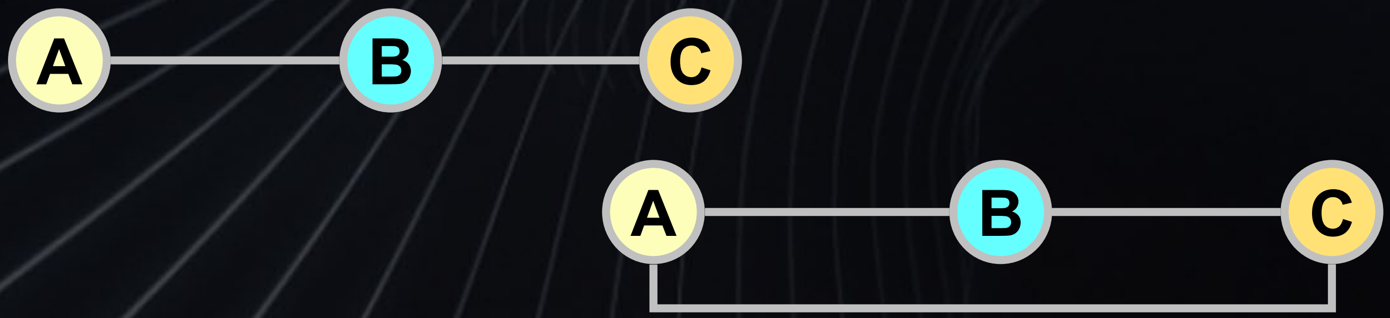
IS B:

Loc (B)	~~DV (A)~~	DV (C)		RT (B)
B, 0	~~A, 0~~	A, inf	⇒	B,loc,0
	~~B, inf~~	B, inf		C,B2 ,1
	~~C, inf~~	C, 0		D,B2, 2
	~~D, inf~~	D, 1		

The Bottom Line

Routers do not know the network topology

Based on distance vectors
B cannot distinguish



Advantages

- → Simple to implement
- → Protocols simple to deploy
  - → Very little configuration

Shortcomings

- Exponential worst case complexity and convergence time
  - $O(n^2)$ to $O(n^3)$ 增加时间复杂度

Shortcomings

- Convergence limited by slower links and routers set pace
- Complex tuning 调试

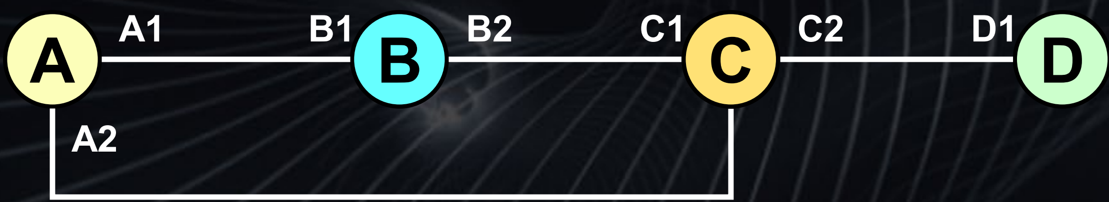
Shortcomings

- → Complex troubleshooting 复杂的故障排除
- → Large routing traffic (and storage) 巨大的路由流量

Not suitable for large complex networks

Path Vector

Eliminates routing loops

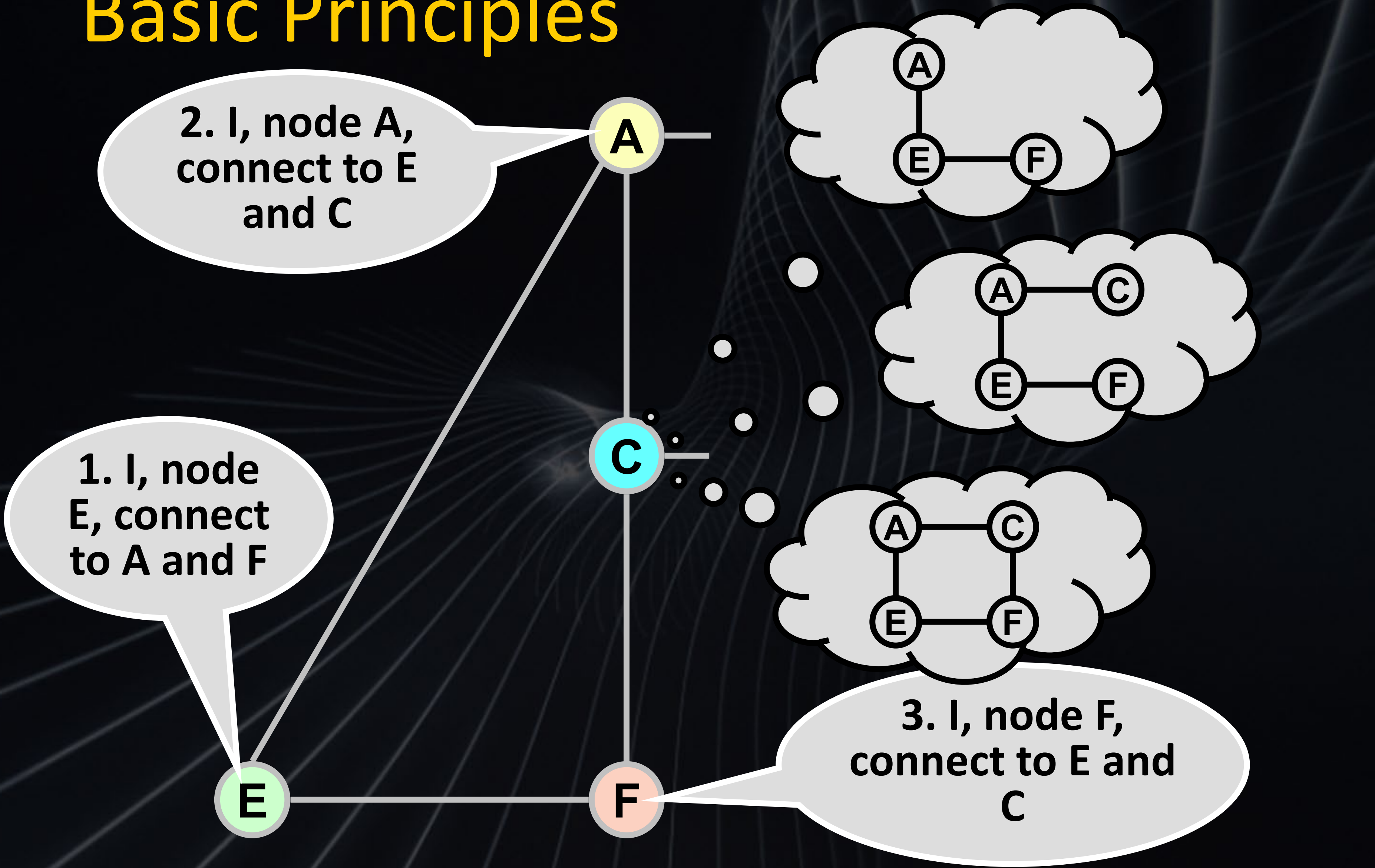


IS A:

Loc (A)	PV (B)	PV (C)	RT (A)	PV (A)
A, 0	A, 1, [B]	A, 1 [C]	A, loc, 0	A, 0, [-]
	B, 0, [-]	B, 1, [B]	B, A1, 1	B, 1, [A]
	C, 1, [B]	C, 0, [-]	C, A2, 1	C, 1, [A]
	D, 2, [B,C]	D, 1, [D]	D, A2, 2	D, 2, [A,C]

LINK STATE ROUTING ALGORITHM

Basic Principles



Basic Principles

- Information on the state of each link
  - Link state
- A local map
- Sent by each node
  to all other nodes
  - Selective flooding

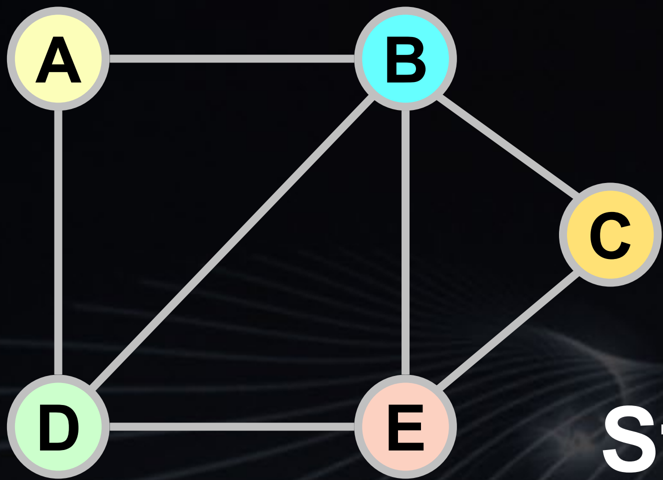
通过有选择的泛洪的方法发给所有节点

Basic Principles

- Nodes build a network map 每个节点都会有网络拓扑图
  - The same on all nodes
- Each node computes routes on the map 每个节点根据这个给图计算路由
  - Dijkstra (shortest path first) algorithm

最短路径优先

Link State Database



Stored by each router

LS (A)	LS (B)	LS (C)	LS (D)	LS (E)
B, 1	A, 1	B, 1	A, 1	B, 1
D, 1	C, 1	E, 1	B, 1	C, 1
	D, 1		E, 1	D, 1
	E, 1			

Dijkstra Algorithm

- Low complexity

低复杂度

  - L • log (N)
    - L: number of links
    - N: number of nodes
- Shortest Path First

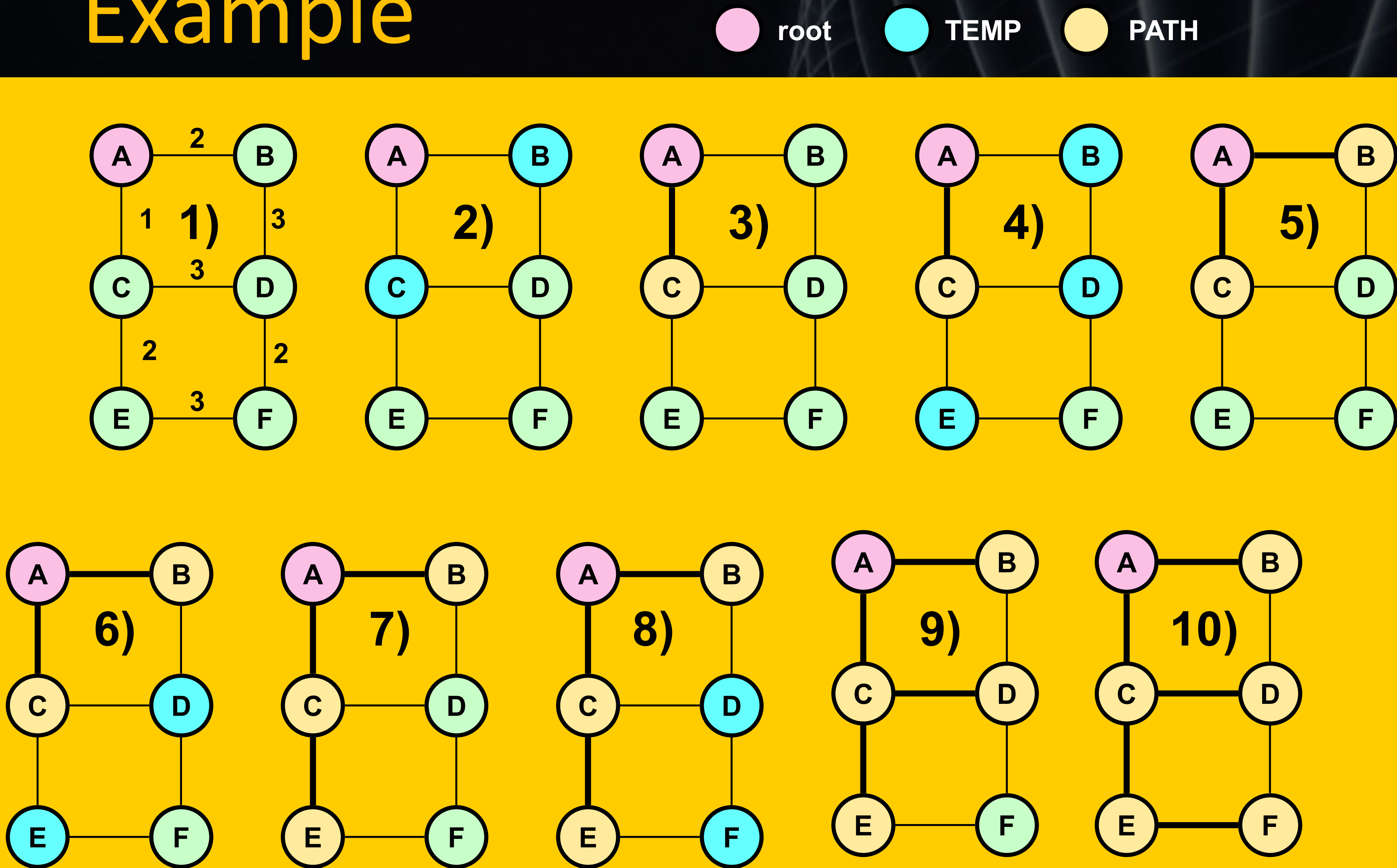
最短路径优先

Shortest Path First

- The next node "nearest" to the root is identified 下一个离根最近的节点会被标识
- Its path is inserted into the routing table

他的路径会被插入到路由表中

Example



Rapid Convergence

快速收斂

link state

- → LSs spread quickly
  - → No intermediate processing

Limited Routing Traffic and Storage

受限于路由流量和存储空间

- → Link states are small
- → Fast and efficient neighbor greeting

快速且高效的邻居问候

其他优点

Other Advantages

- It rarely generates loops
- Simple to understand and troubleshoot
    - All nodes have identical databases

Shortcoming

很高的部署复杂度

- High implementation complexity
  - Selective flooding 选择性泛洪
  - First implementation took several years 第一次部署需要花几年时间
- Protocols with complex configuration

协议要求复杂的配置

Link State Generation 链路状态的生成

- In principle: when there is a topology change 当拓扑结构改变的时候
- Actual protocols: LS are generated periodically LS是周期性生成的
  - Increased reliability 增加了可靠性