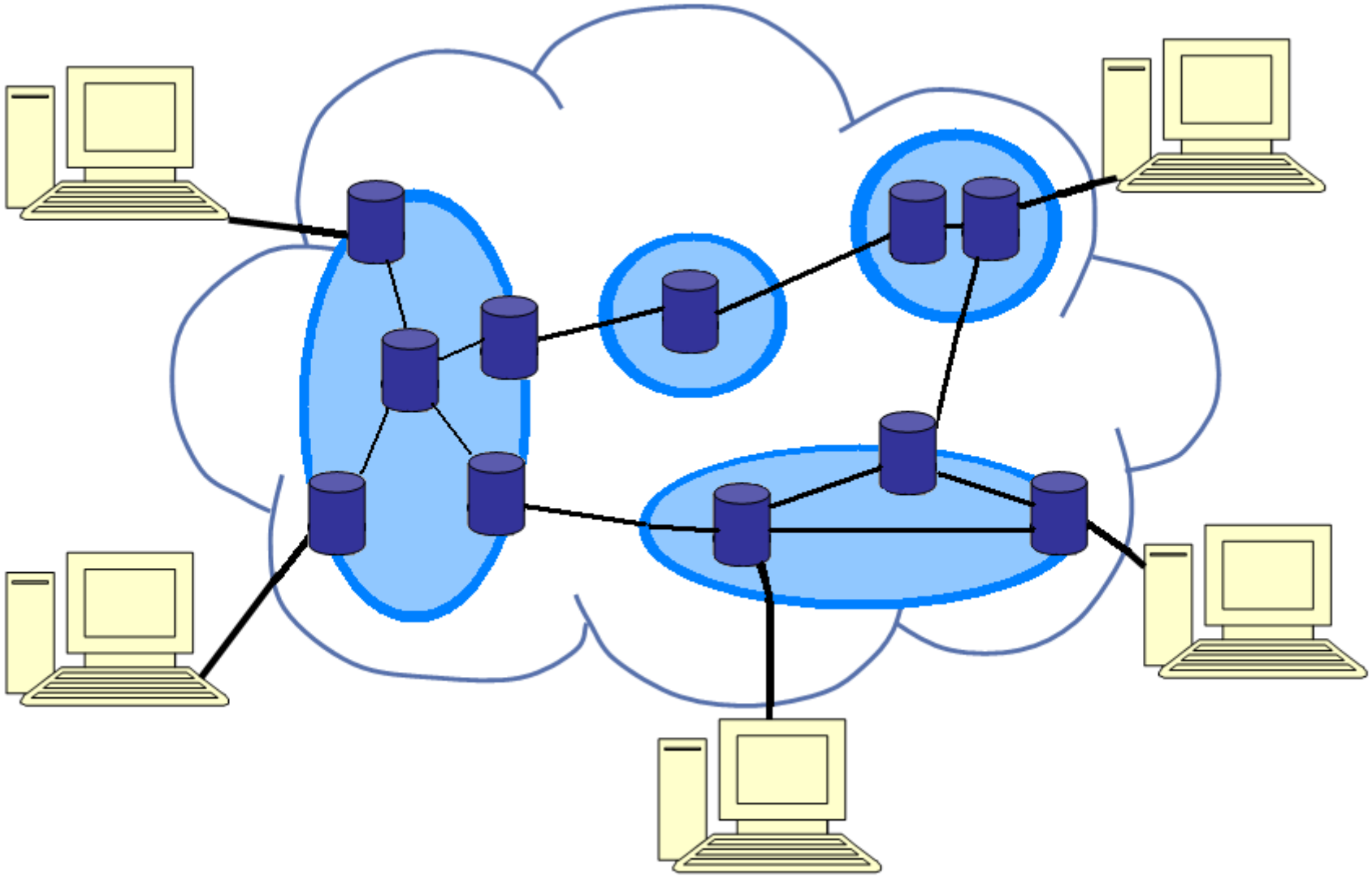
# Using Forgetful Routing to Control BGP Table Size

Elliott Karpilovsky, Jennifer Rexford
Computer Science Department, Princeton University

# The Internet: A Router Level View



# WorldCom suffers widespread Internet outage

*USA Today, 2005*

# Comcast Internet outages lead to nationwide frustration among customers

*Denver Post, 2005*

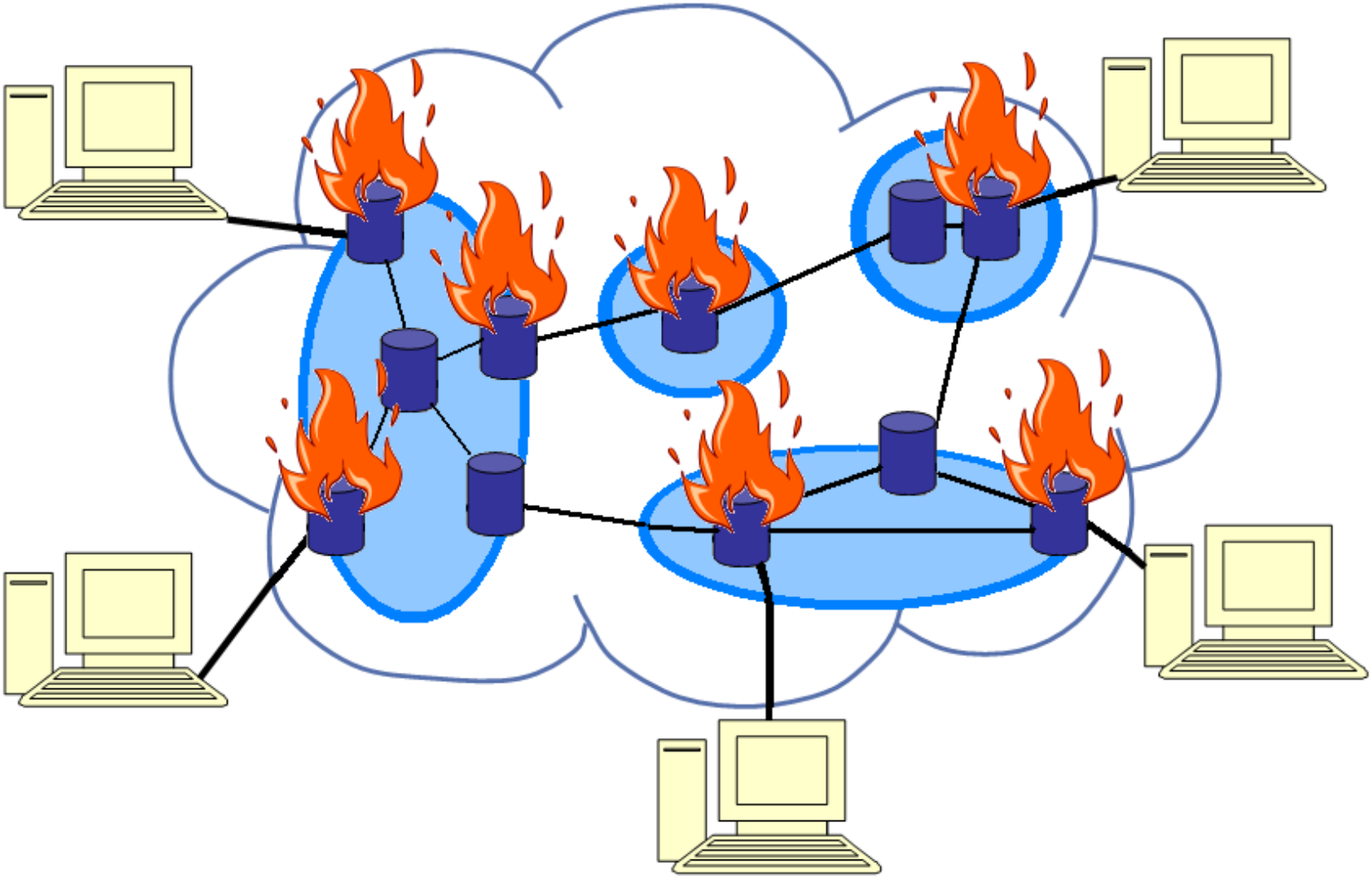
# Tier1 ISPs Dying

*Slashdot, 2005*

# U.S. unprepared for Net meltdown

*News.com, 2006*

# The Internet: A Router Level View



# Source of Routing Failures

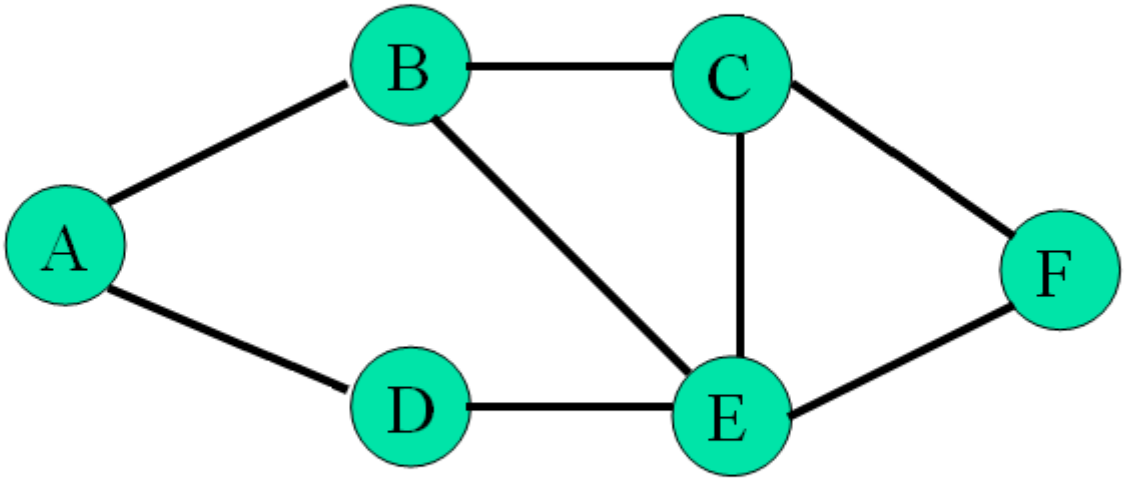
- Memory
- Memory
- Memory

# Overview

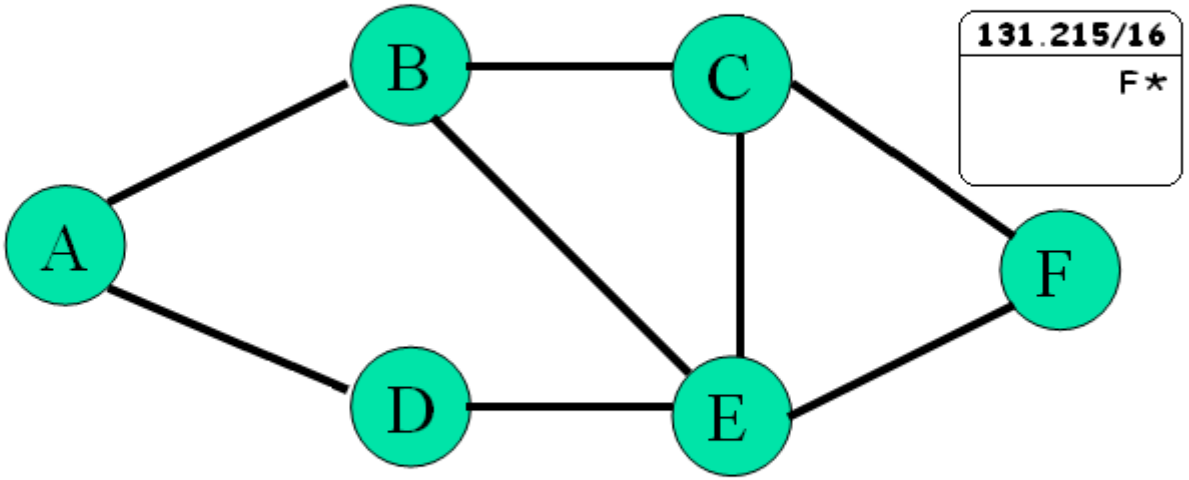
- Internet Routing (and its Memory Issues)
- Current Solutions (and Their Problems)
- Forgetful Routing (the Theory and Application)
- Future Directions (and Conclusion)

# Internet Routing

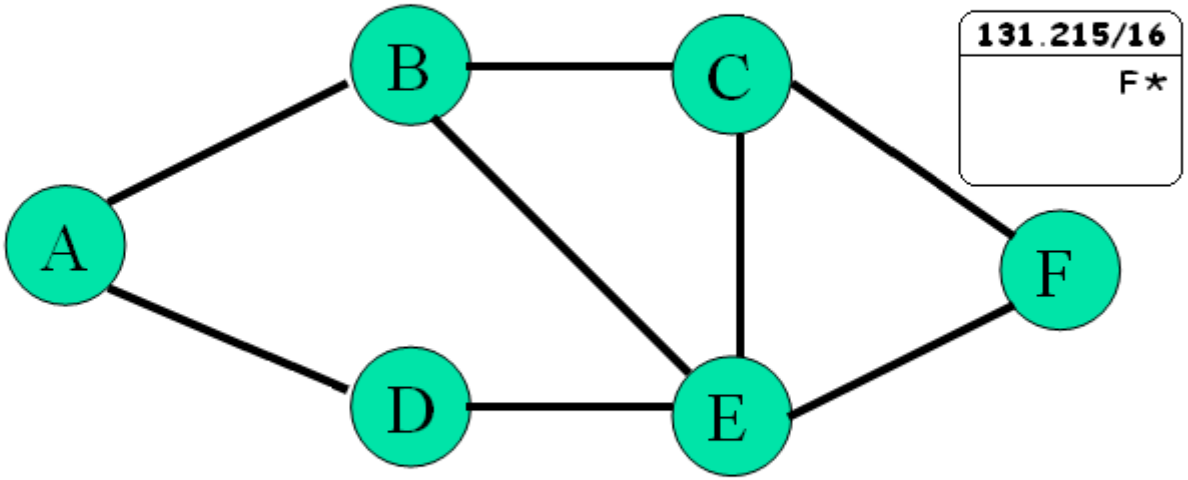
# How BGP Works



# How BGP Works

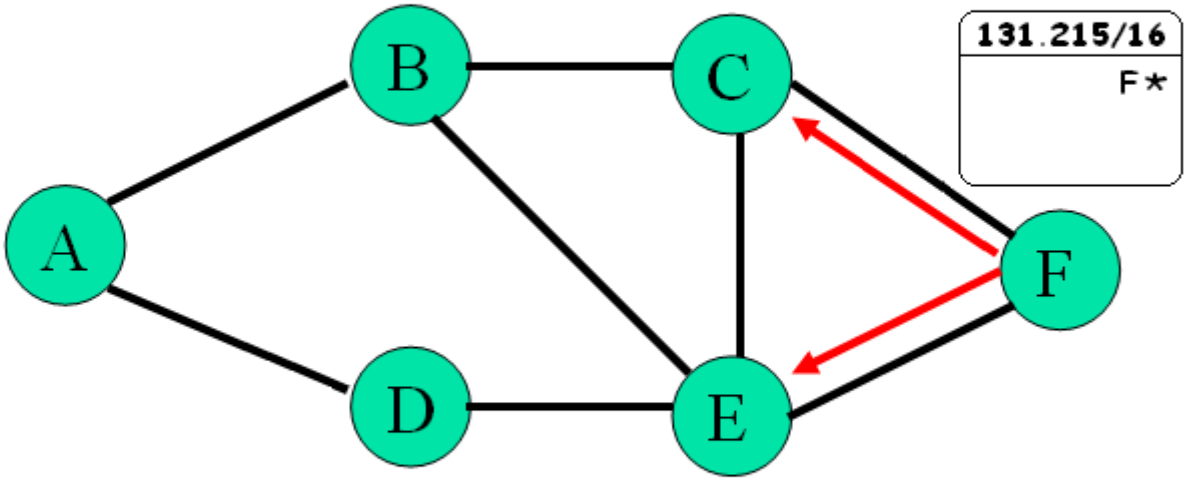


# How BGP Works

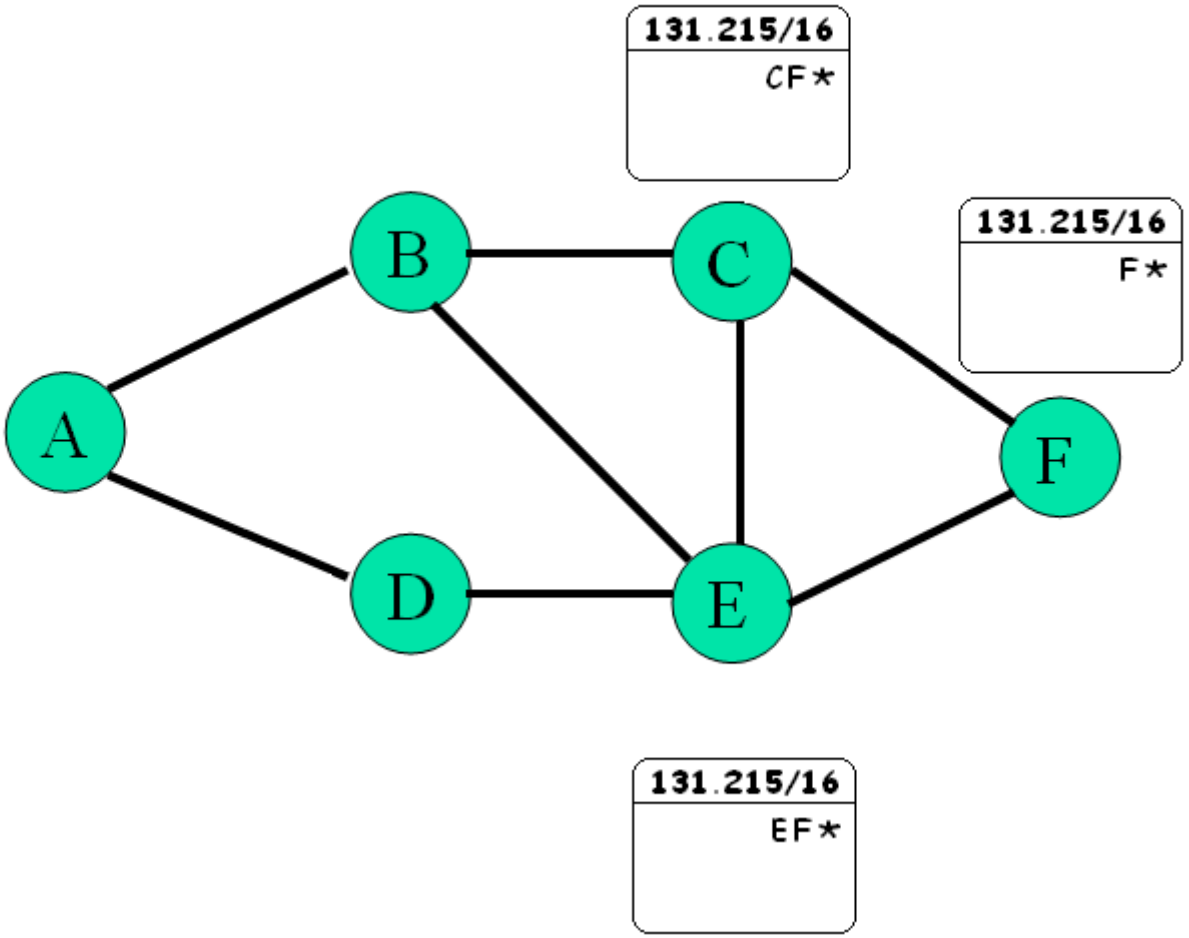



Prefix based

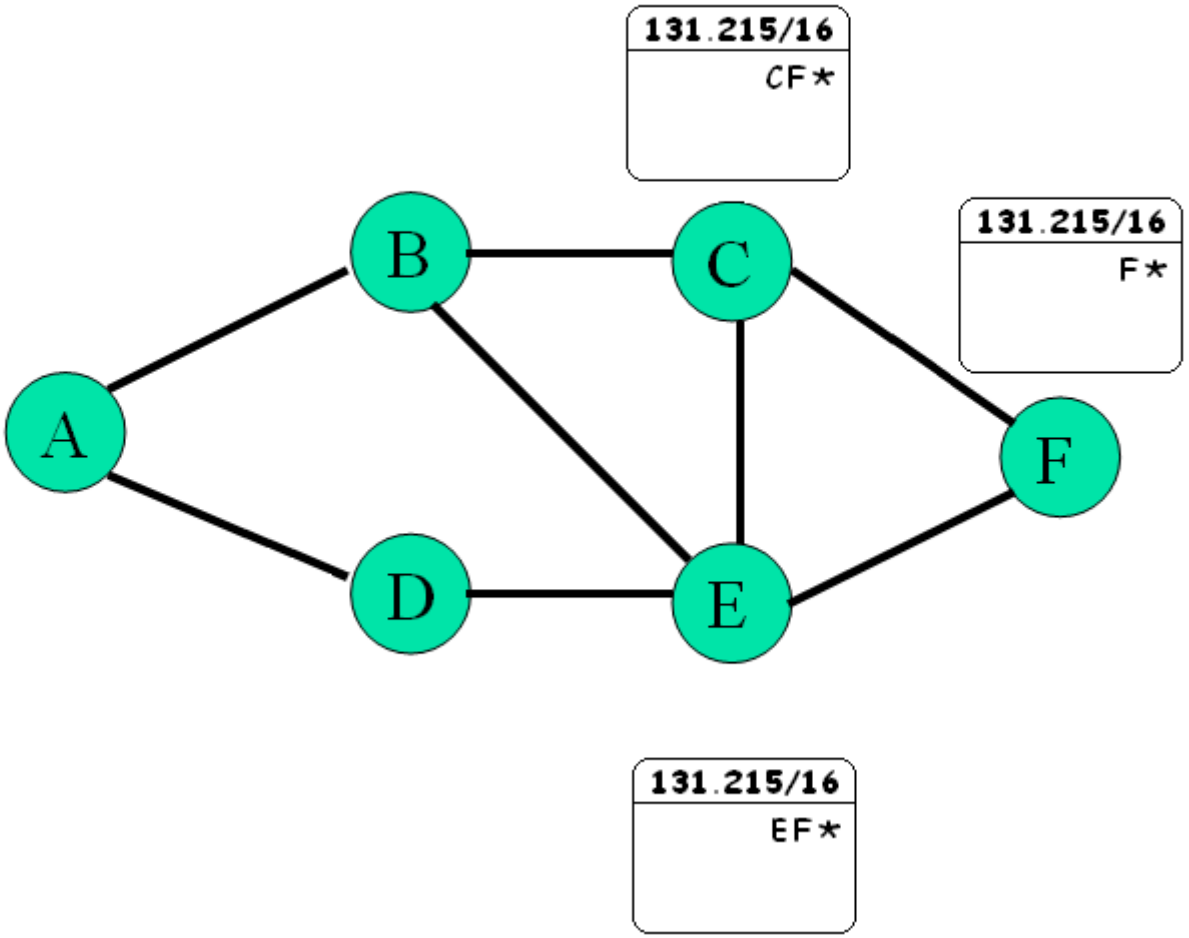
# How BGP Works



# How BGP Works

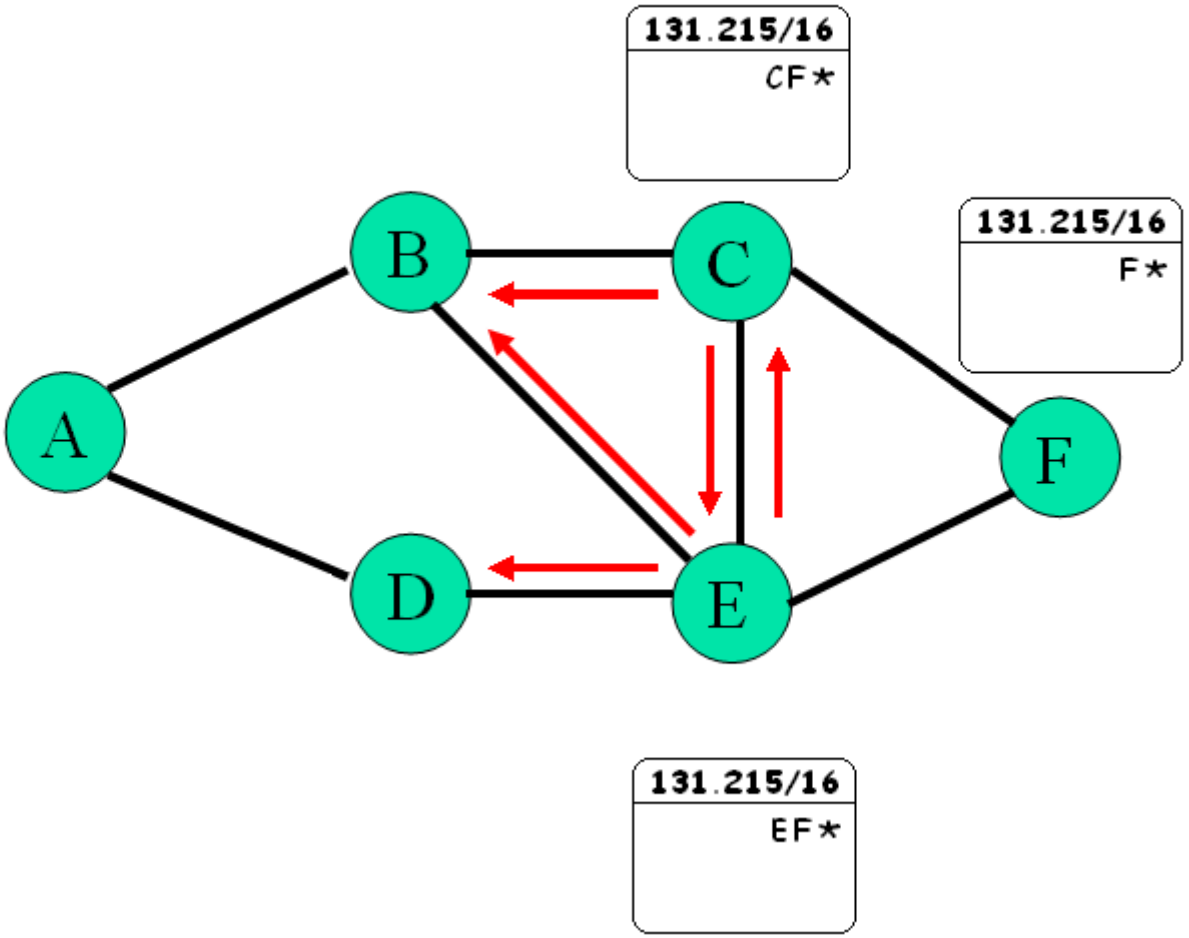


# How BGP Works

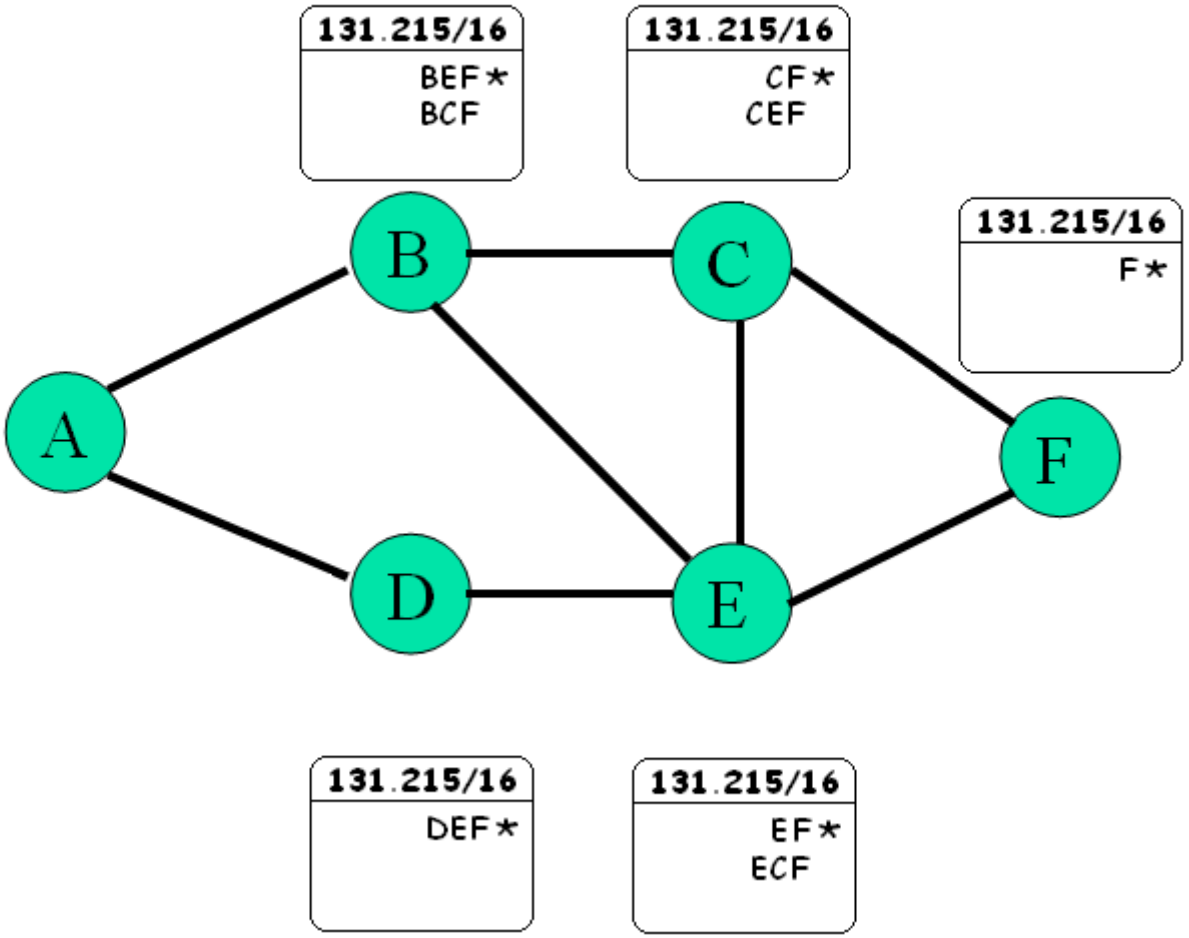



Path vector protocol

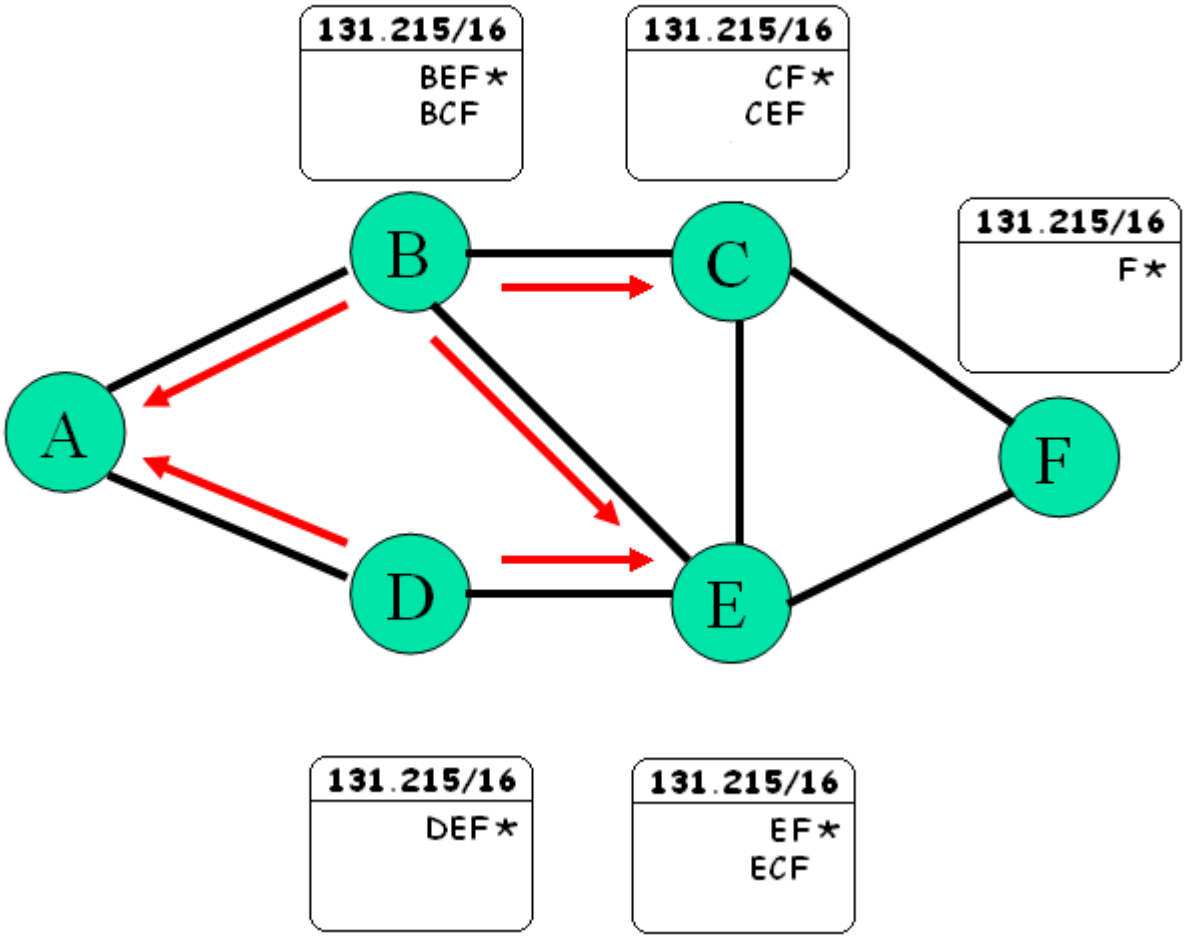
# How BGP Works



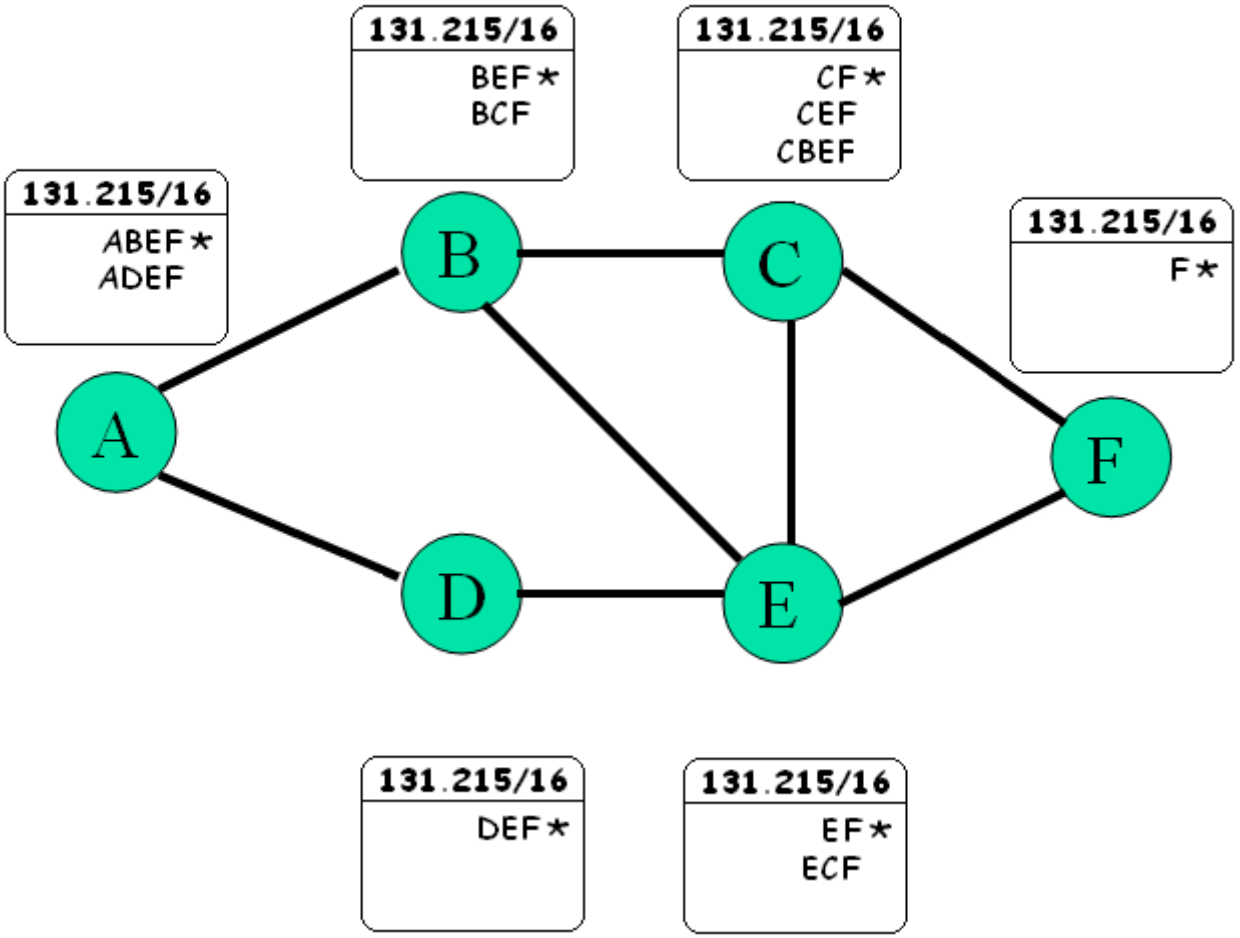
# How BGP Works



# How BGP Works

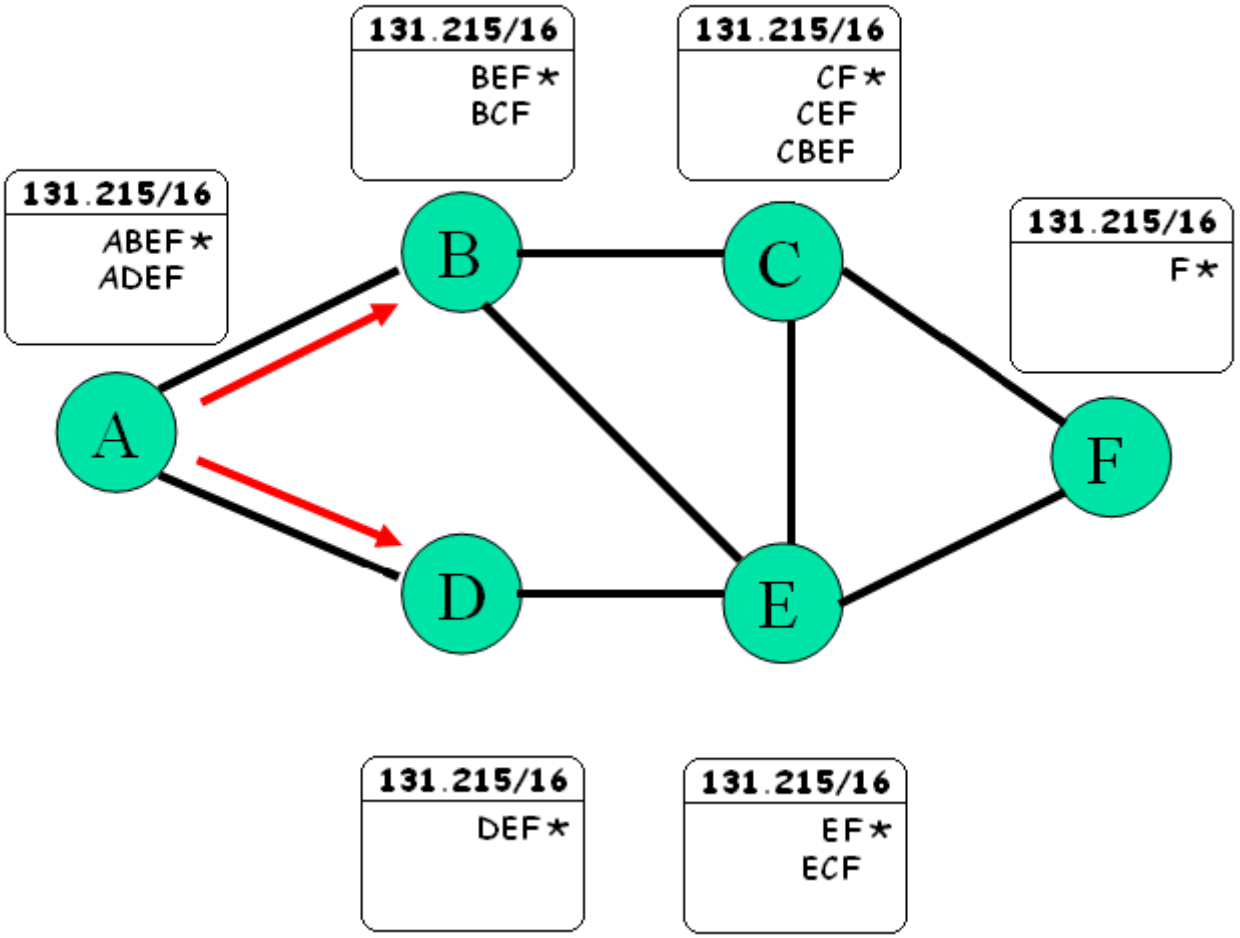


# How BGP Works

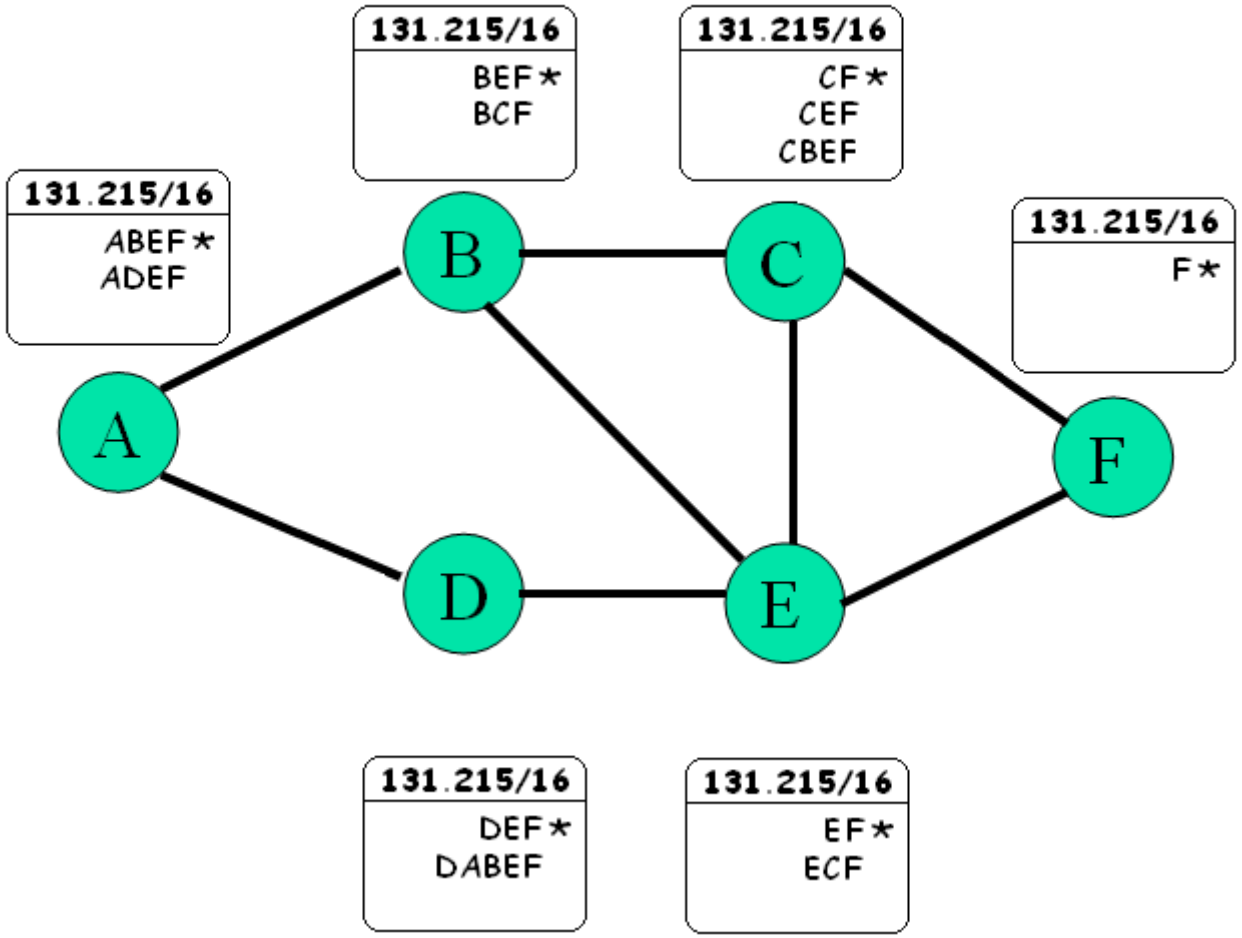



Only best route is propagated

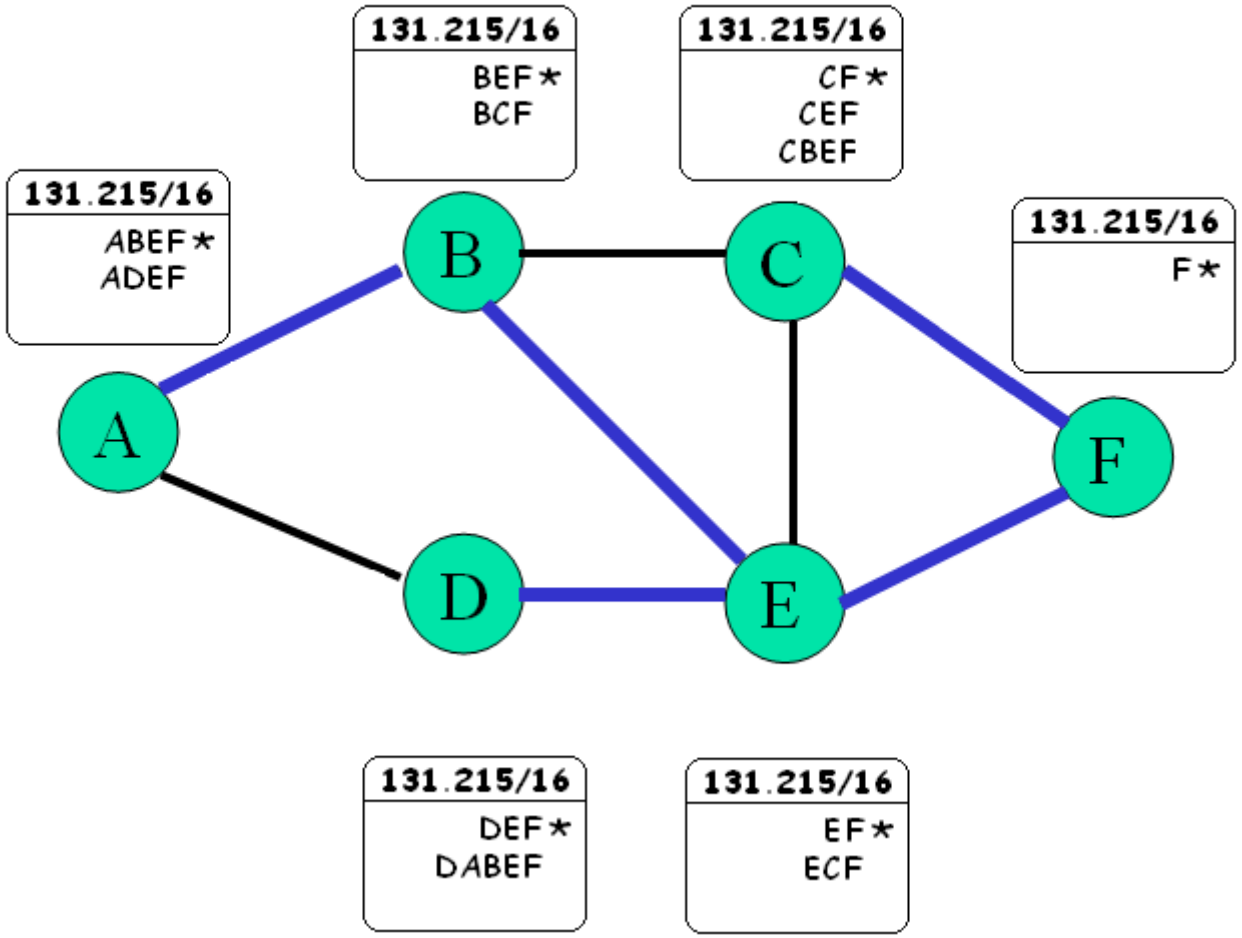
# How BGP Works



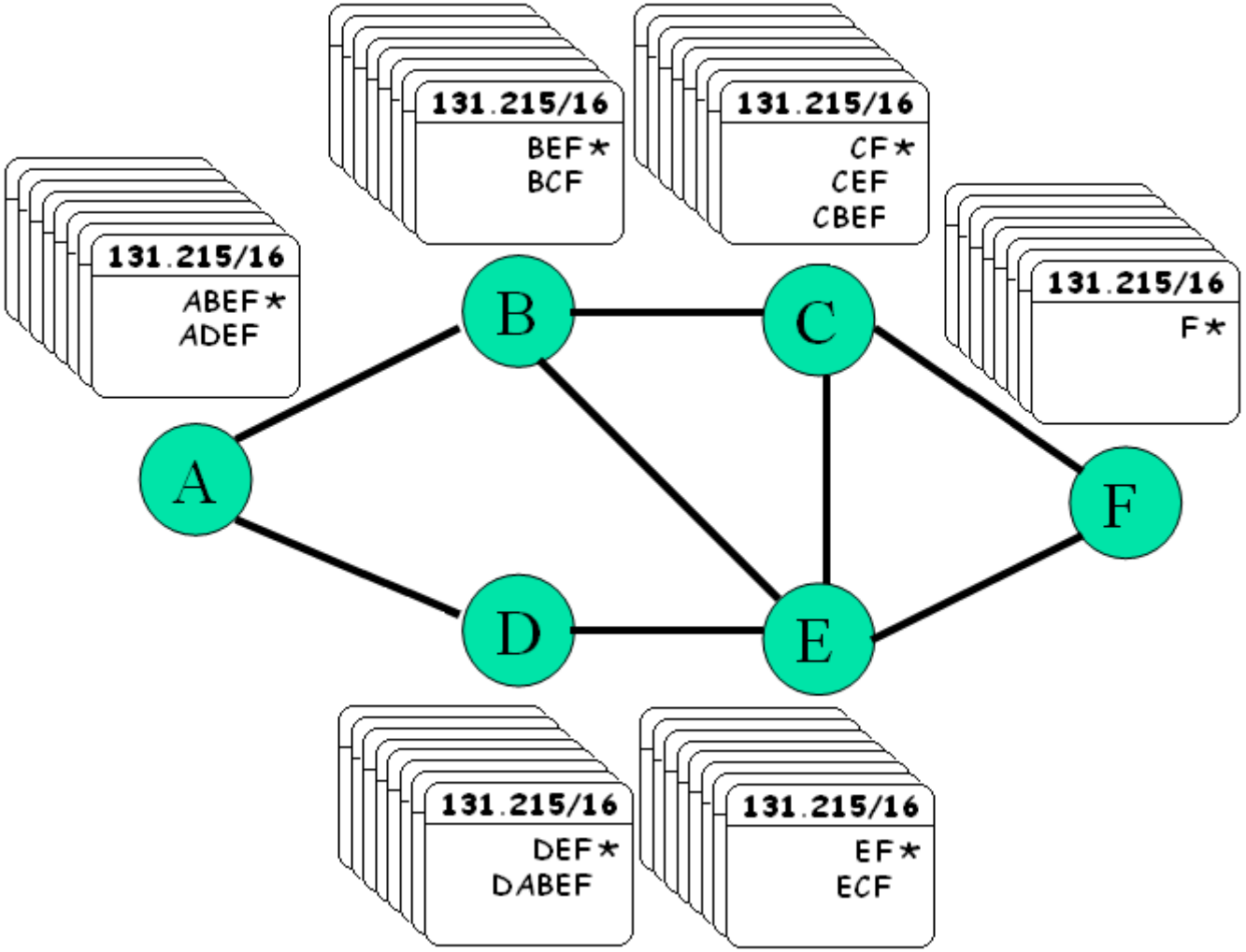
# How BGP Works



# How BGP Works



# How BGP Works



# Router Memory Problems

- Many, many prefixes
  - Approximately 170,000 prefixes currently in use
- Path diversity
  - Many routes may be learned per prefix
- Path vector protocol
  - Stores entire path for each route
- Incremental protocol
  - Once a route is learned, it is not re-advertised

# Router Memory

- Routing Information Base (RIB)
  - Maps prefixes to lists of possible routes
  - Stored in memory on the router
  - Can grow very large in size
  - Focus of our research is RIB reduction
- Forwarding Information Base (FIB)
  - Maps prefixes to their next-hop
  - Stored in line cards
  - Optimal memory reduction techniques exist

# Current Solutions

# Memory, a big deal?

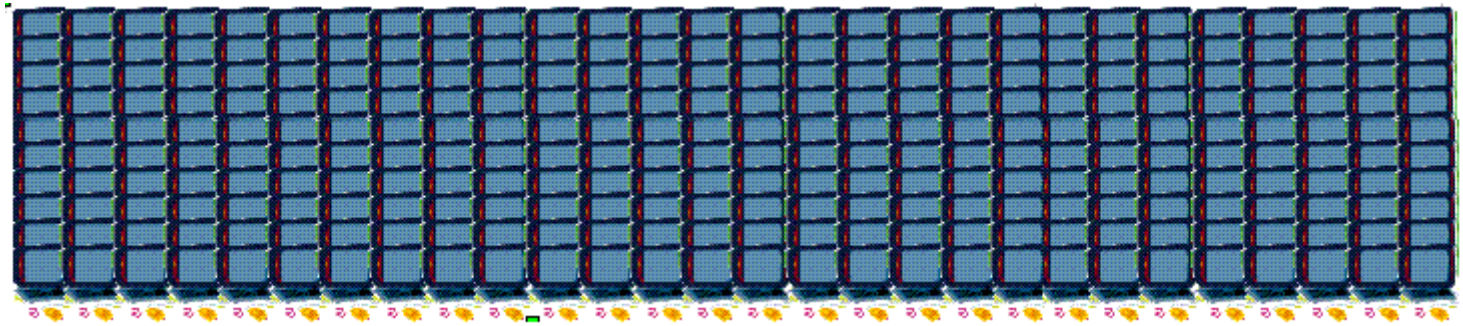
# Memory, a big deal

- Cost

# Memory, a big deal...

- Cost

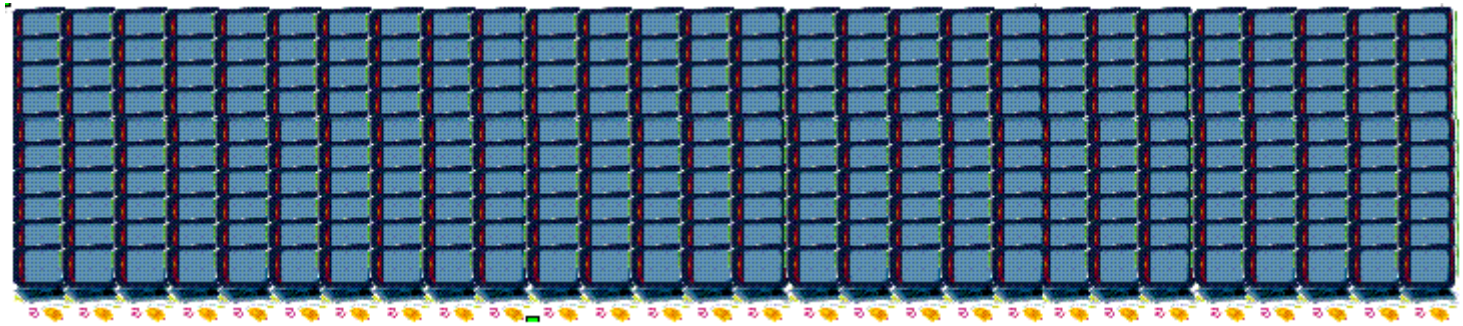

- Quantity



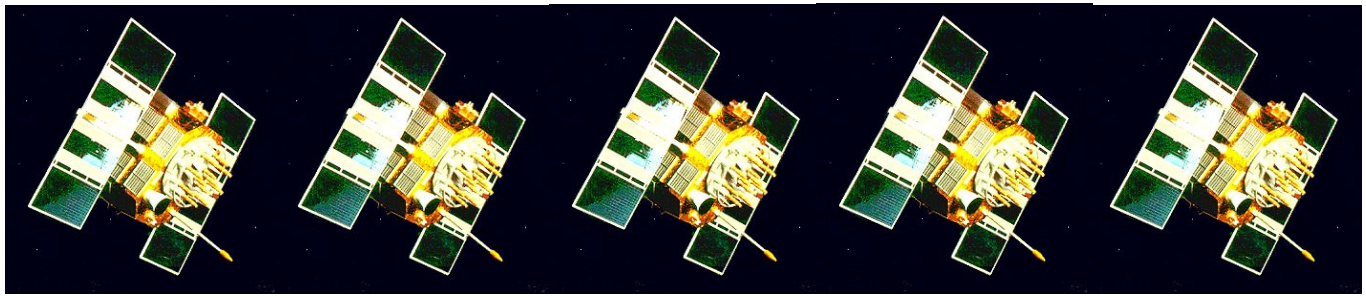
# Memory, a big deal!

- Cost


- Quantity



- Installation
  (image courtesy of NASA)



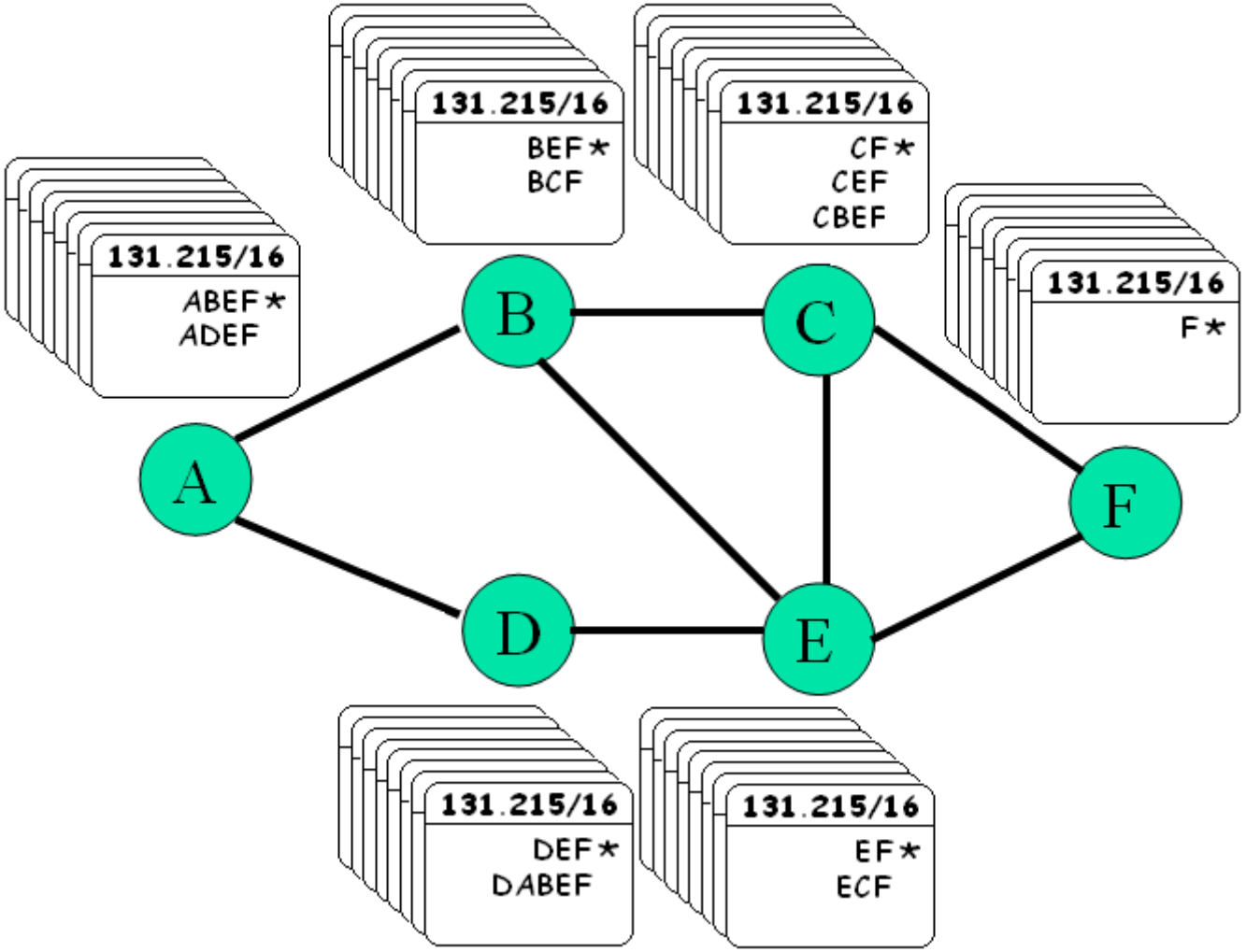
# Operator Solutions

- Filter unexpected prefixes
  - But is everything known ahead of time?
- Prefix limits
  - But what about connectivity?
- Guidelines to filter small address blocks
  - But what about ISPs that don't follow guidelines?

# BGP is the problem... so change BGP!

- New architectures proposed to replace BGP
  - Tunneling to core routers, aggressive aggregation, etc.
- Big problem: not incrementally deployable
  - No “flag day” for the Internet to switch over to a new protocol

# What to do?

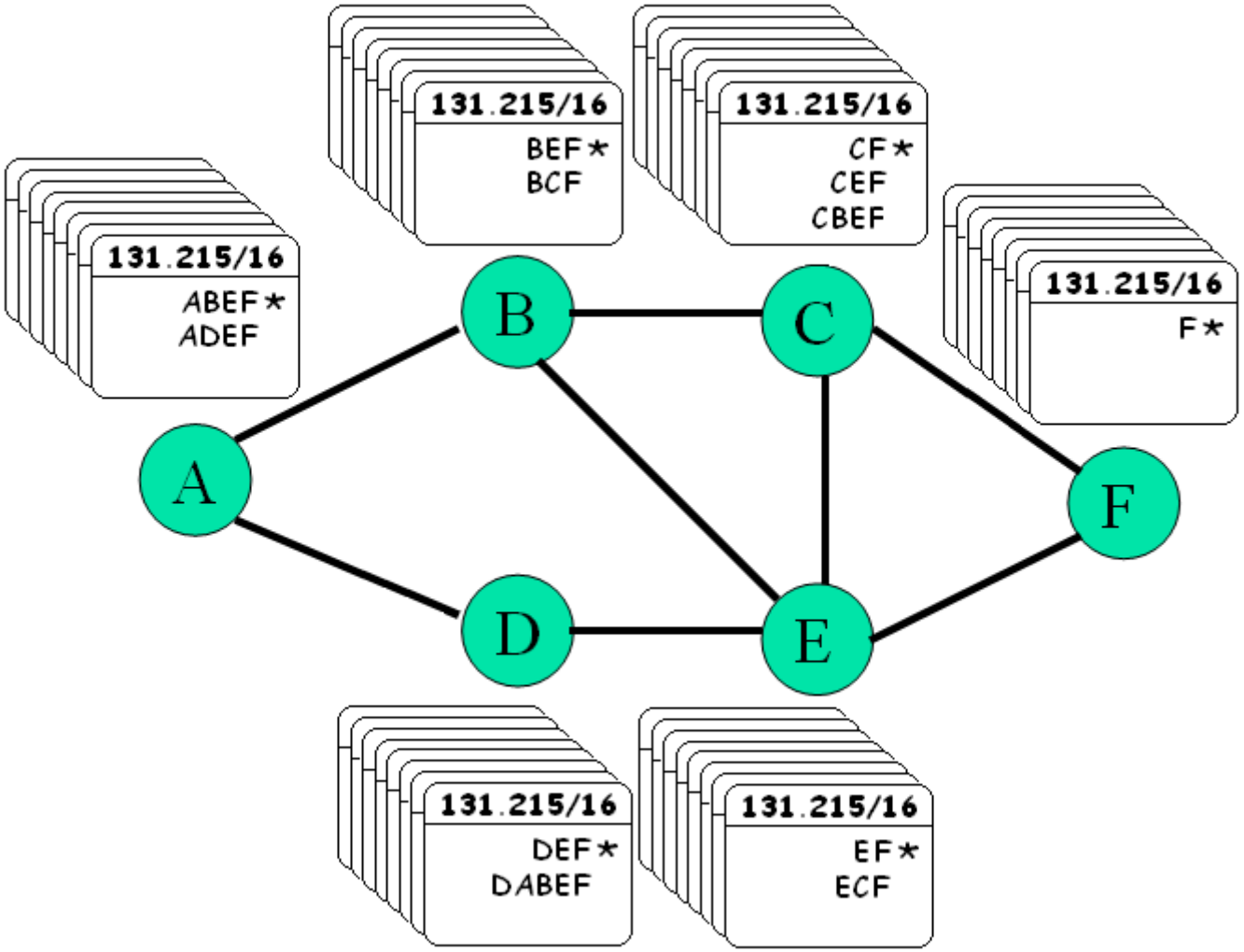

131.215/16
ABEF*
ADEF
131.215/16
BEF*
BCF
131.215/16
CF*
CEF
CBEF
131.215/16
F*
A
B
C
D
E
F
131.215/16
DEF*
DABEF
131.215/16
EF*
ECF

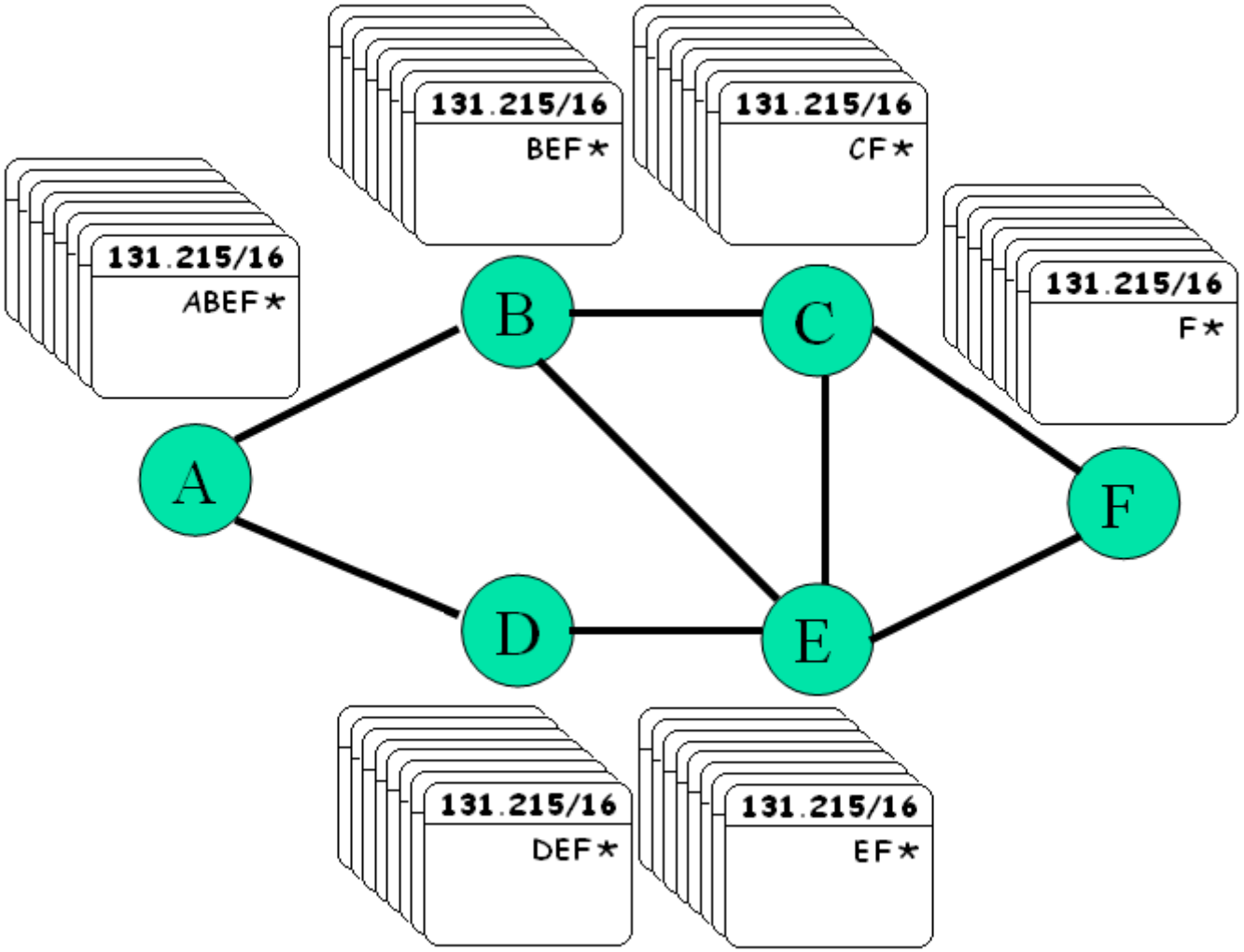
# Forgetful Routing

# Key Insight

Every secondary route is some other router's primary route

If every router always remembers its primary, all routing information can be reconstructed


131.215/16
ABEF*
ADEF
131.215/16
BEF*
BCF
131.215/16
CF*
CEF
CBEF
131.215/16
F*
A
B
C
D
E
F
131.215/16
DEF*
DABEF
131.215/16
EF*
ECF


131.215/16
ABEF*
131.215/16
BEF*
131.215/16
CF*
131.215/16
F*
A
B
C
D
E
F
131.215/16
DEF*
131.215/16
EF*

# RFC 2918 – Route Refresh

- Allows a BGP speaker to send a “refresh” message to neighbor
- Neighbors receiving this message re-advertise their outbound routes
- Supported on all modern CISCO and Juniper routers

# A “Cache Replacement” Problem

- Exchange possible bandwidth usage for memory savings

- When low on memory, what do we evict?
  - Will affect the number of refreshes needed later

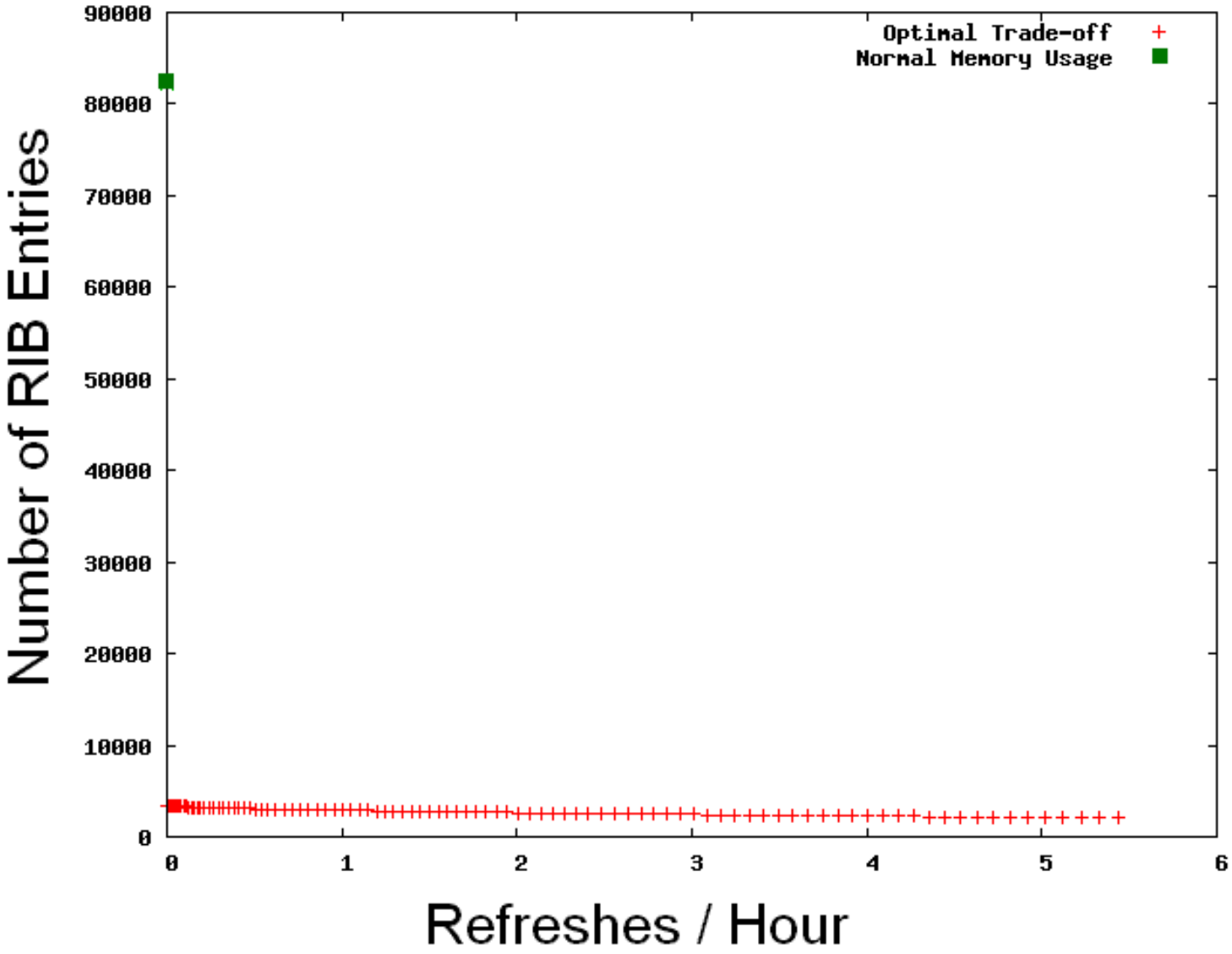
# Theoretical Limits

- Using foresight, look into the future
- Identify alternate routes that are never needed
- Identify alternate routes that are needed furtherest in the future
- For simplicity, treat RIB entries as fixed length

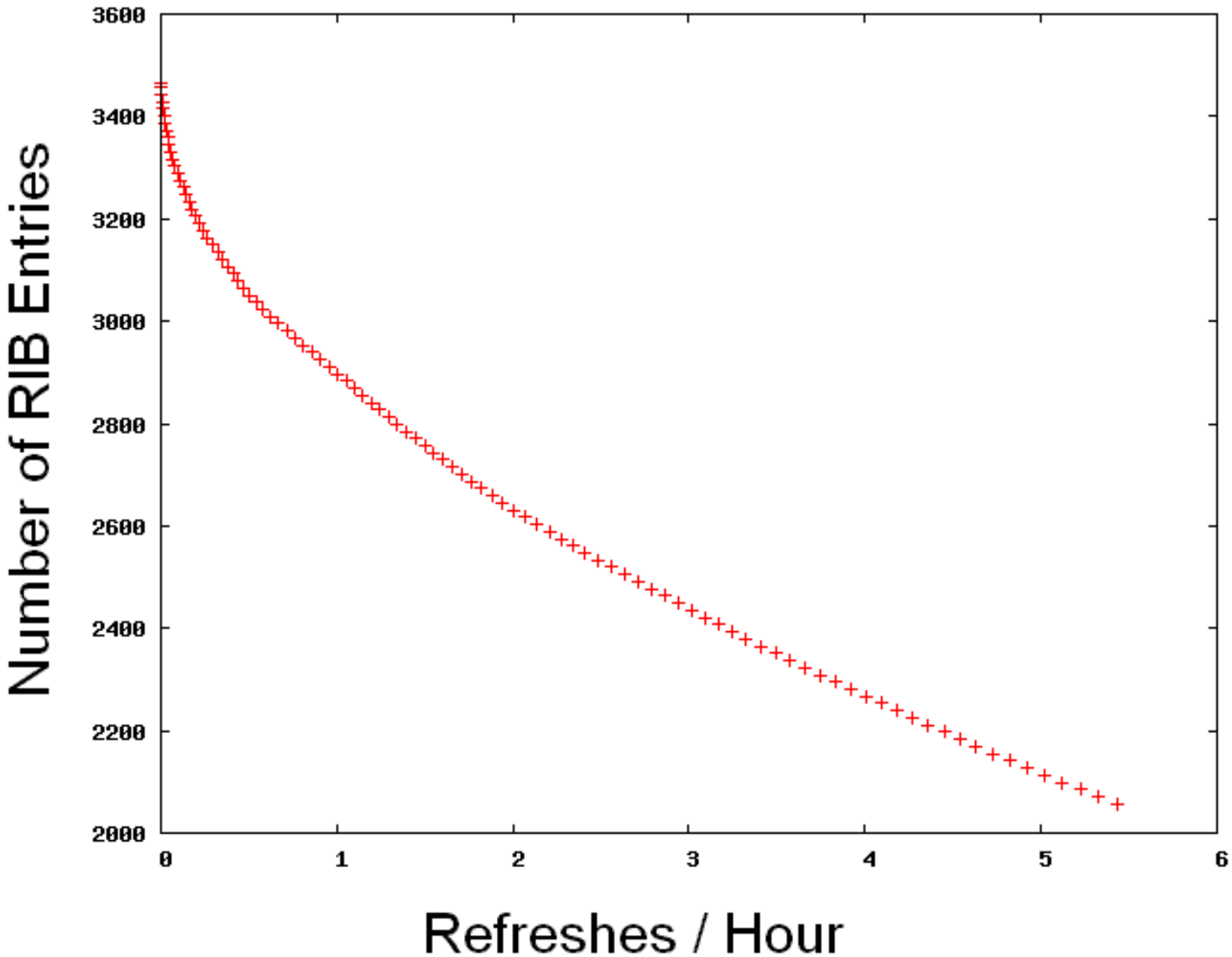
# Evaluation

- Used RouteViews data from 2005/01/01 to 2005/07/01, sampled at 1%
  - Approximately 2000 prefixes
- Created an optimal, offline algorithm to establish a lower bound

# Optimal Trade-off Curve



# Optimal Trade-off Curve



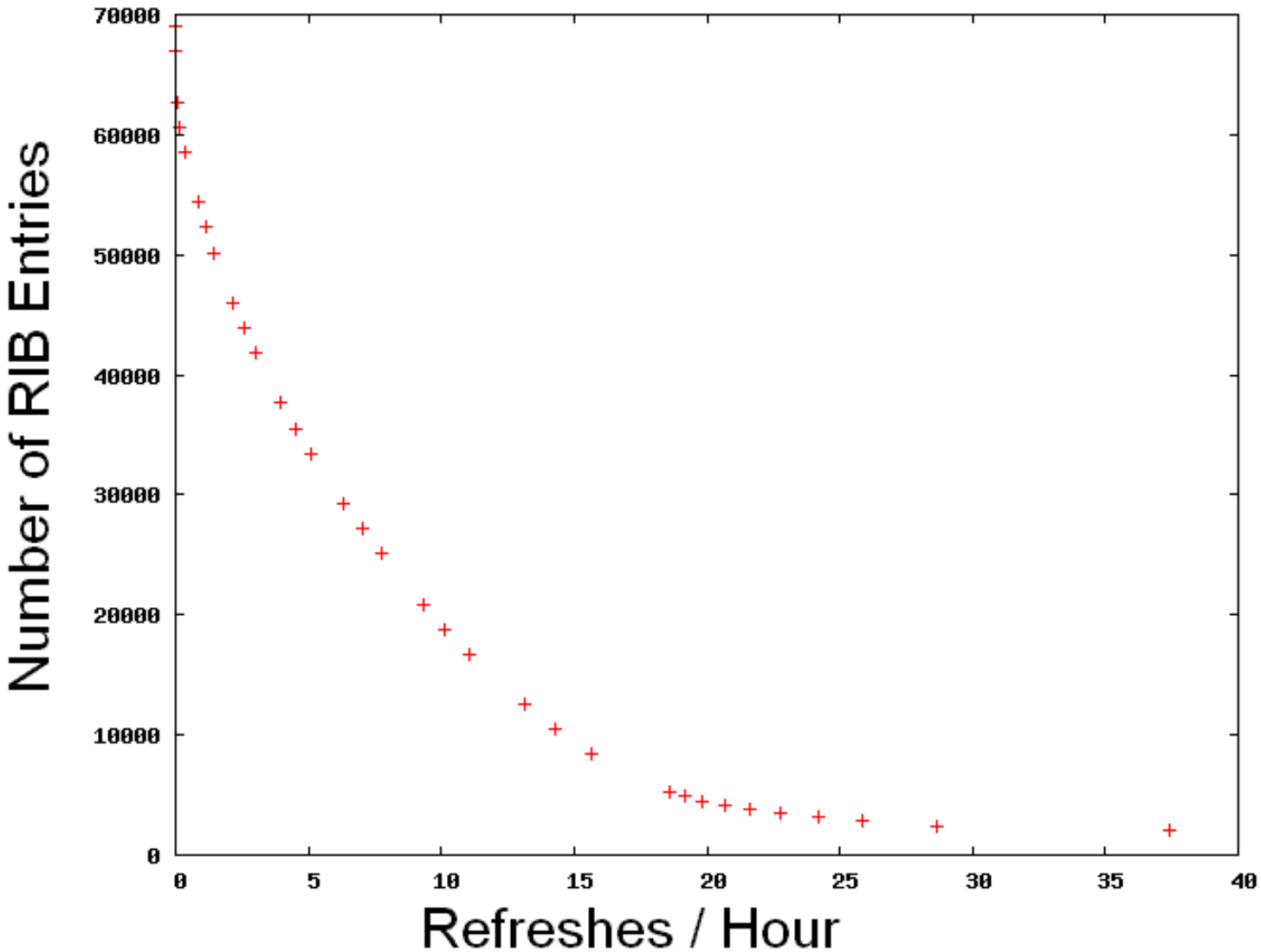
# Evaluation

- Used RouteViews data from 2005/01/01 to 2005/07/01, sampled at 1%
  - Approximately 2000 prefixes
- Created several different online algorithms and compared their results
  - Constraints: O(1) time overhead, minimal space overhead

# Algorithm: Least Recently Used

- Routes are ordered by time since they were last used as a primary route
- Maintain a doubly-linked list in memory for O(1) time overhead

# Algorithm: Least Recently Used



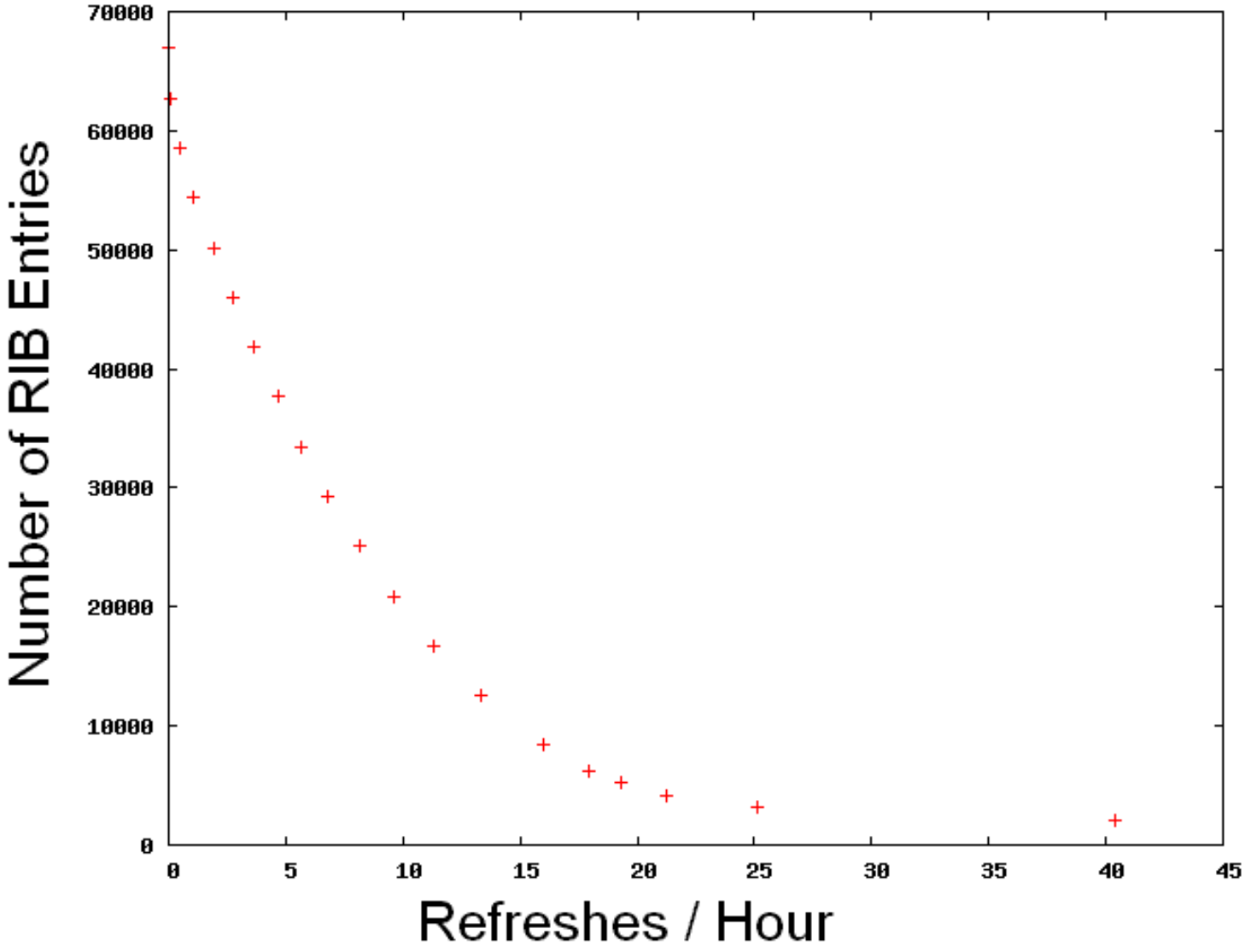
# Algorithm: Least Recently Used

- Good performance
- Potentially bad memory overhead
  - An additional 8 bytes per route
  - Will consume anywhere from 1.4 megabytes to 40 megabytes of memory in practice

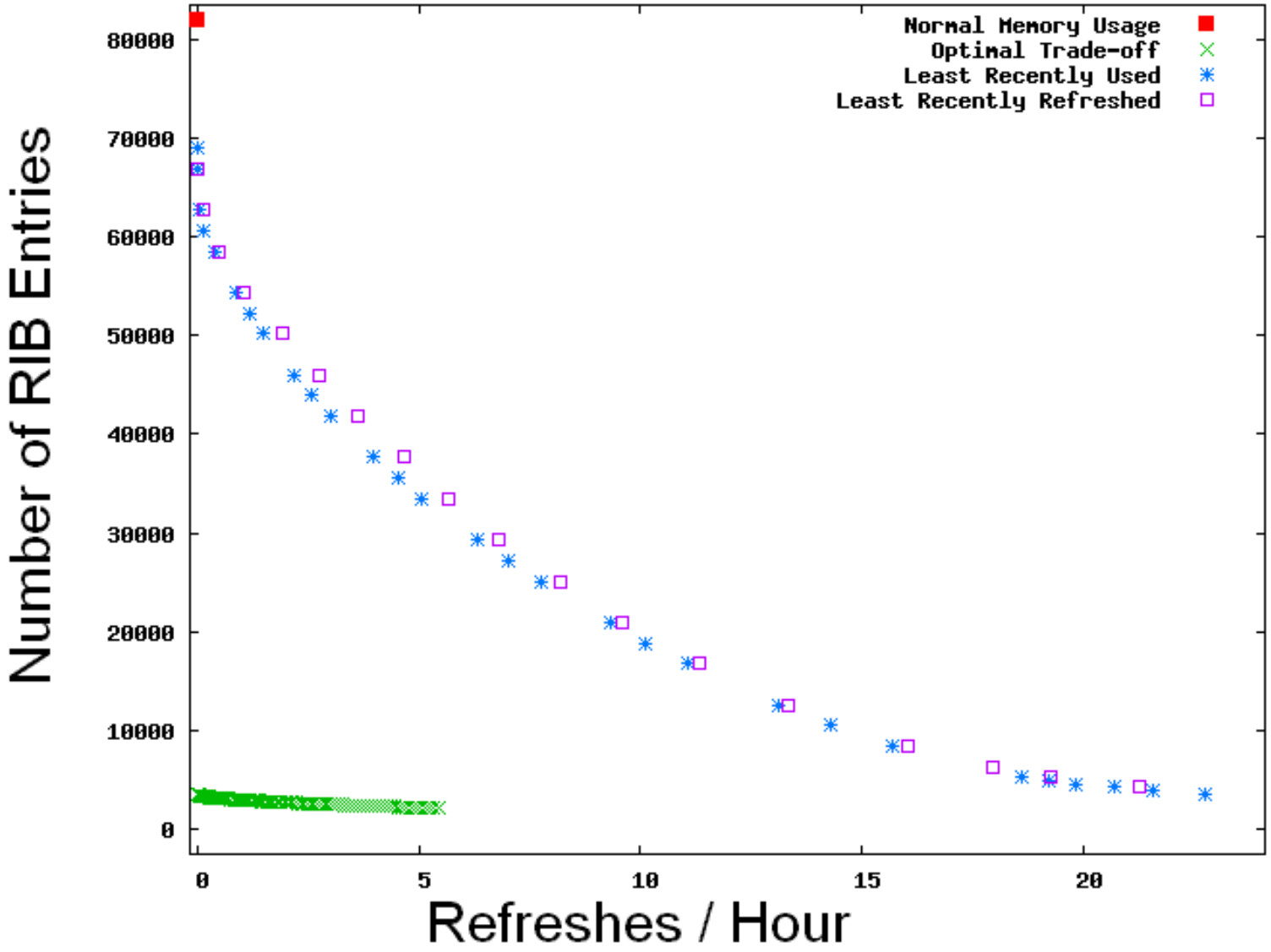
# Algorithm: Least Recently Refreshed

- Prefixes are ordered by time since they last needed a refresh
  - The least preferred route from the most stable prefix is evicted
- Maintain a doubly-linked list in memory for O(1) time overhead
- Memory overhead is now 8 bytes per prefix
  - Will consume about 1.4 megabytes of memory in practice

# Algorithm: Least Recently Refreshed



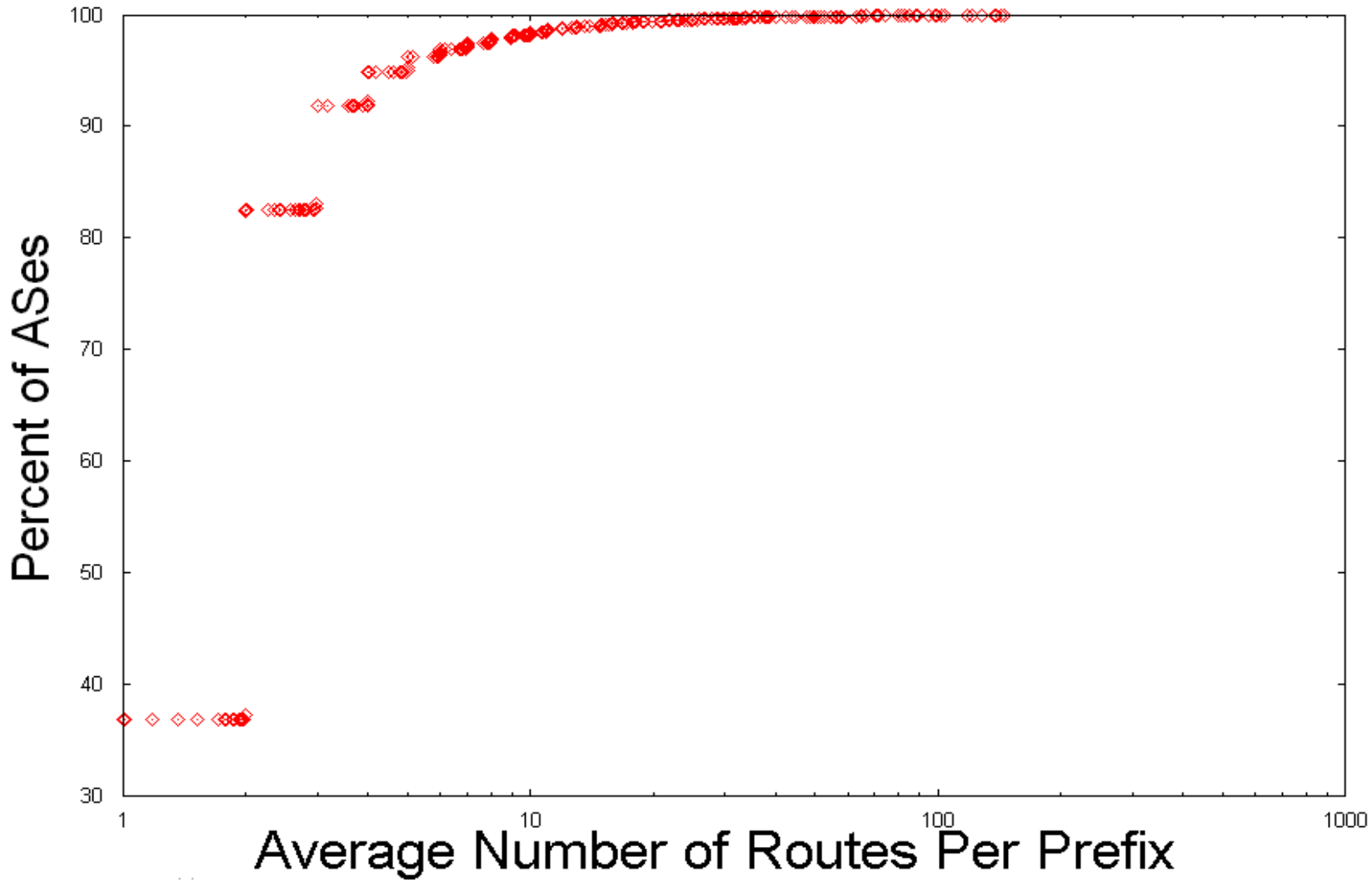
# Comparison


Normal Memory Usage
Optimal Trade-off
Least Recently Used
Least Recently Refreshed
Number of RIB Entries
80000
70000
60000
50000
40000
30000
20000
10000
0
0
5
10
15
20
Refreshes / Hour

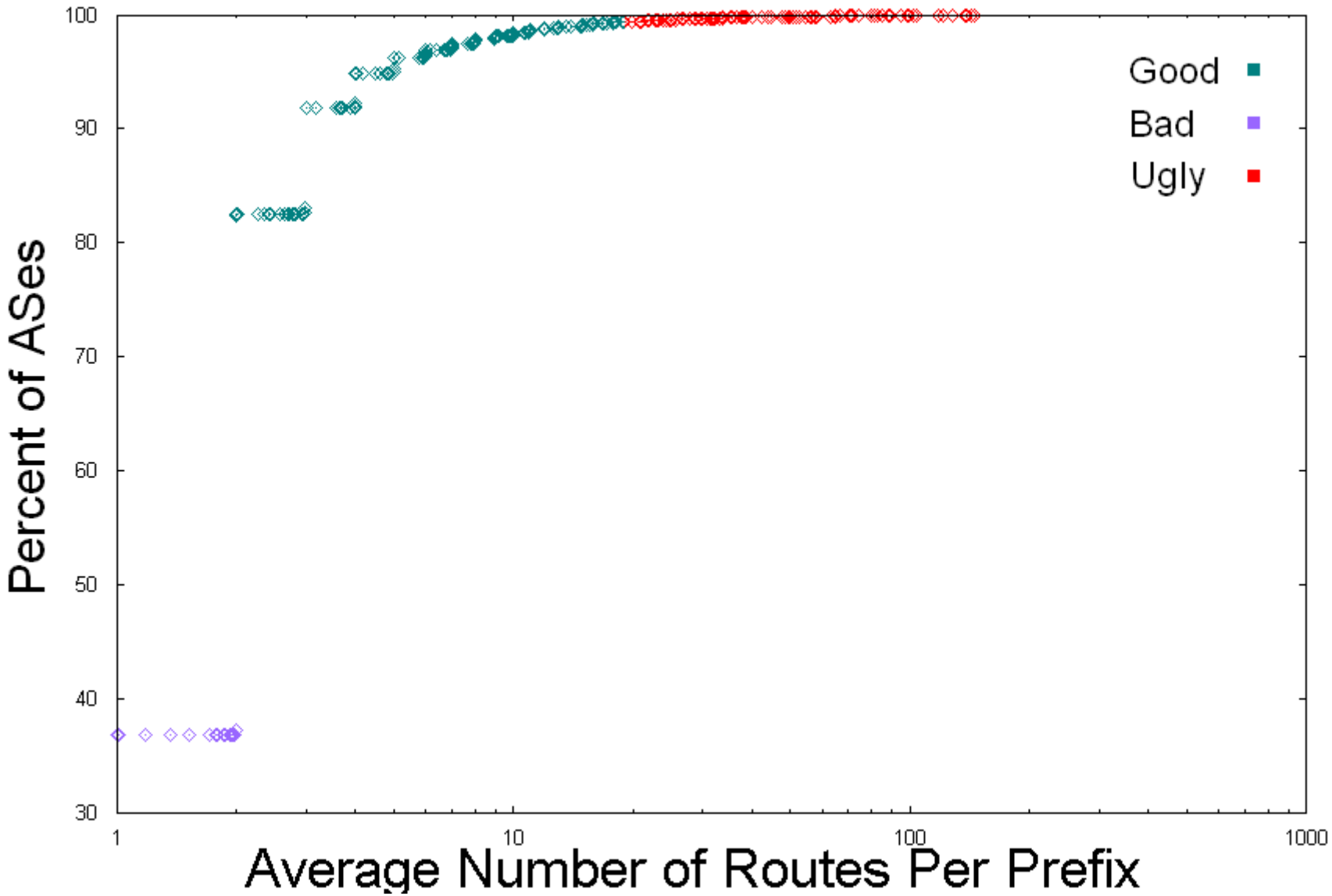
# Issue – Do we have Alternates?

- Do real routers have alternate routes that can be evicted?
  - Depends on the router

# CDF of Avg. Num. Of Routes Per Prefix, Gao Inference on RouteViews, 2005/2/10



# CDF of Avg. Num. Of Routes Per Prefix, Gao Inference on RouteViews, 2005/2/10



# Future Directions

# Future Directions

- Can we create better online algorithms?
- Can we acquire a better data source than RouteViews?
- How would such a system perform connected to a real network?
- Can we reduce the number of prefixes needed without causing “bad” behavior?

# Thank you to

- Nick Feamster
- Changhoon Kim
- Anja Feldmann
- Randy Bush
- Wen Xu
- Anonymous reviewers