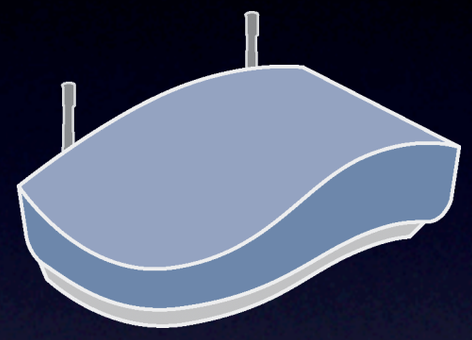
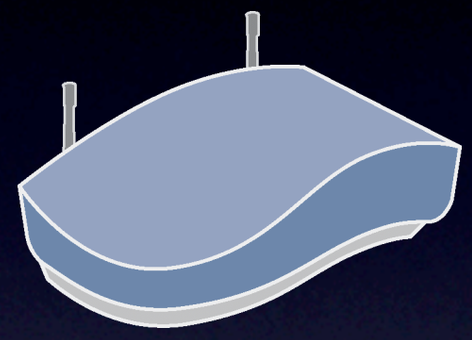
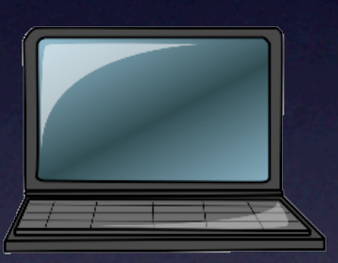
# Wireless network measurement challenges

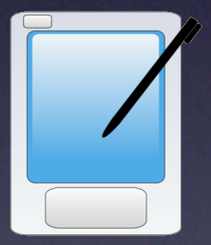
David Kotz

Department of Computer Science
Institute for Security Technology Studies
Dartmouth College

# Lots of wireless devices...

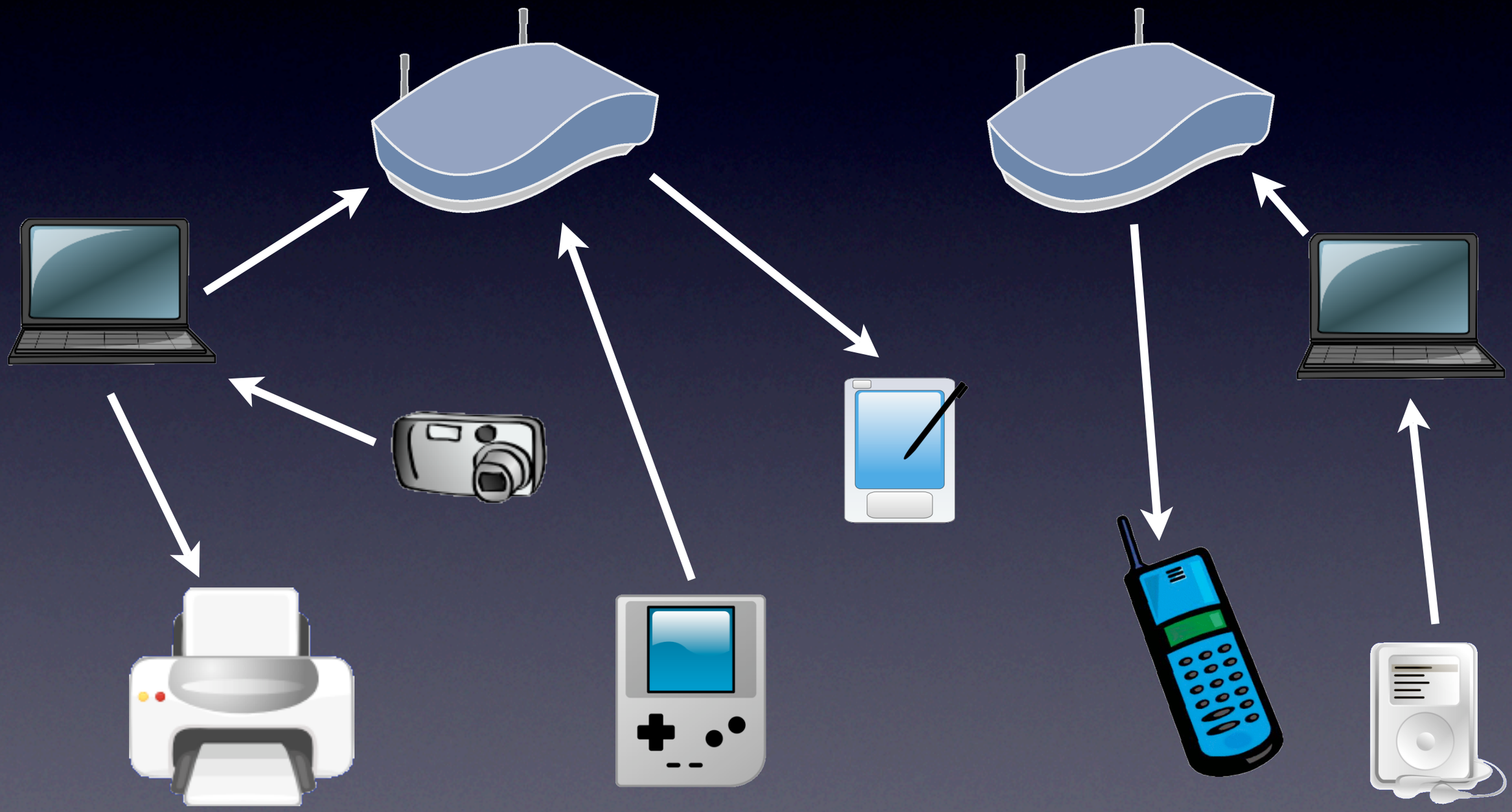







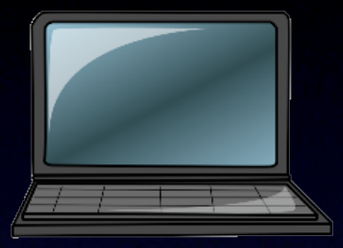


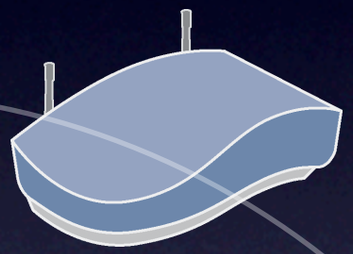
= lots of wireless traffic...

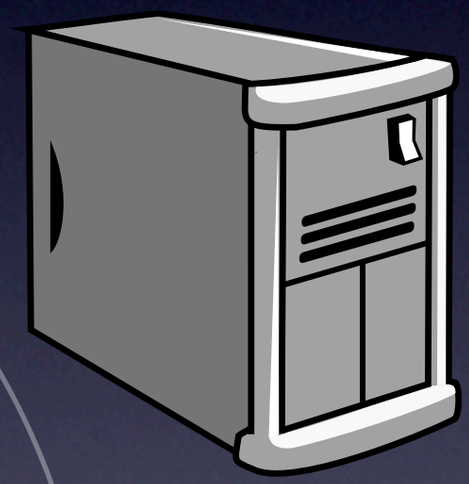


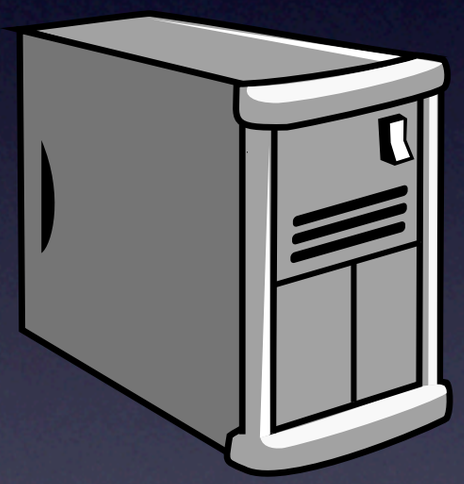
# Why measure?

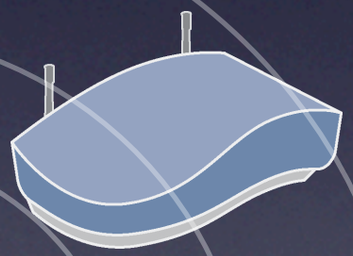
- Operational
  - observe misbehaving/misconfigured/cheating users
  - capture and detect 802.11 MAC-layer attacks
  - capacity planning, trouble shooting
- Research
  - understand devices: VoIP phones, gaming devices, ...
  - develop better MAC protocols and mobility models
- Note: wireless is not wired!
  - new usage patterns, more mobility, different connectivity

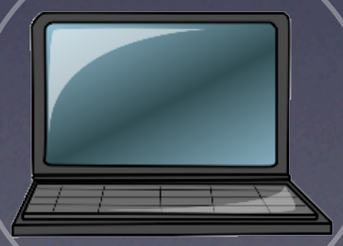




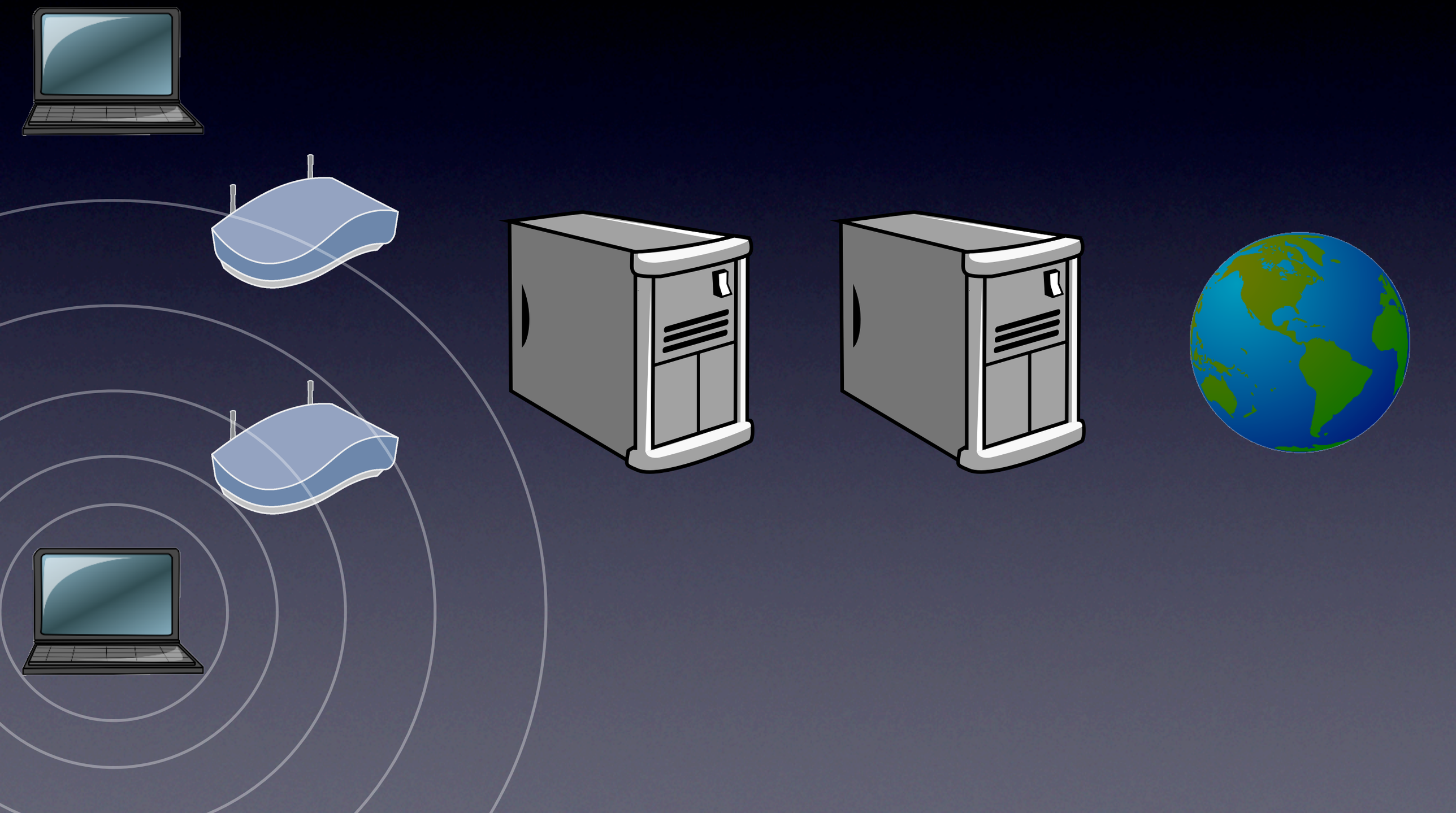




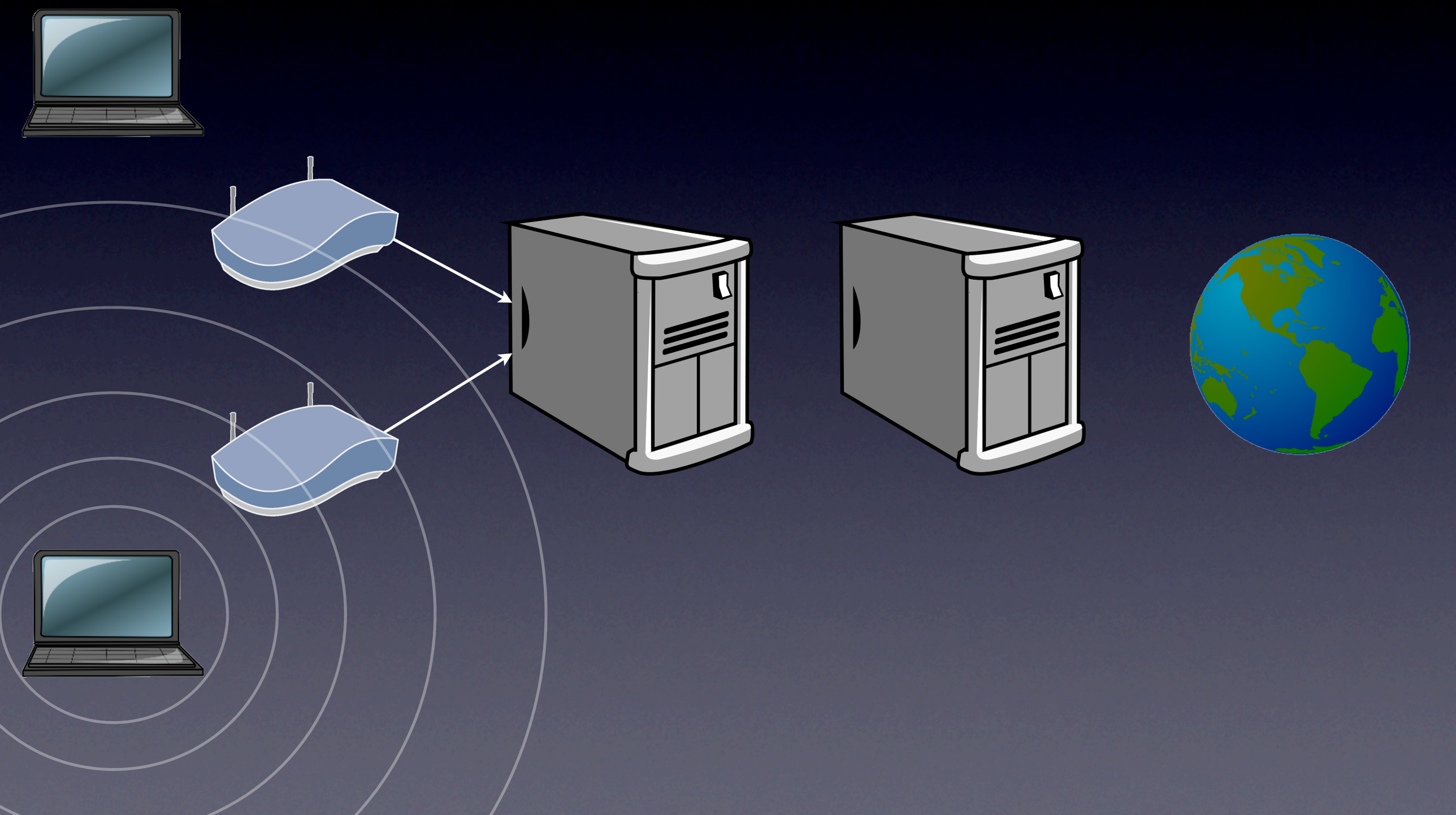




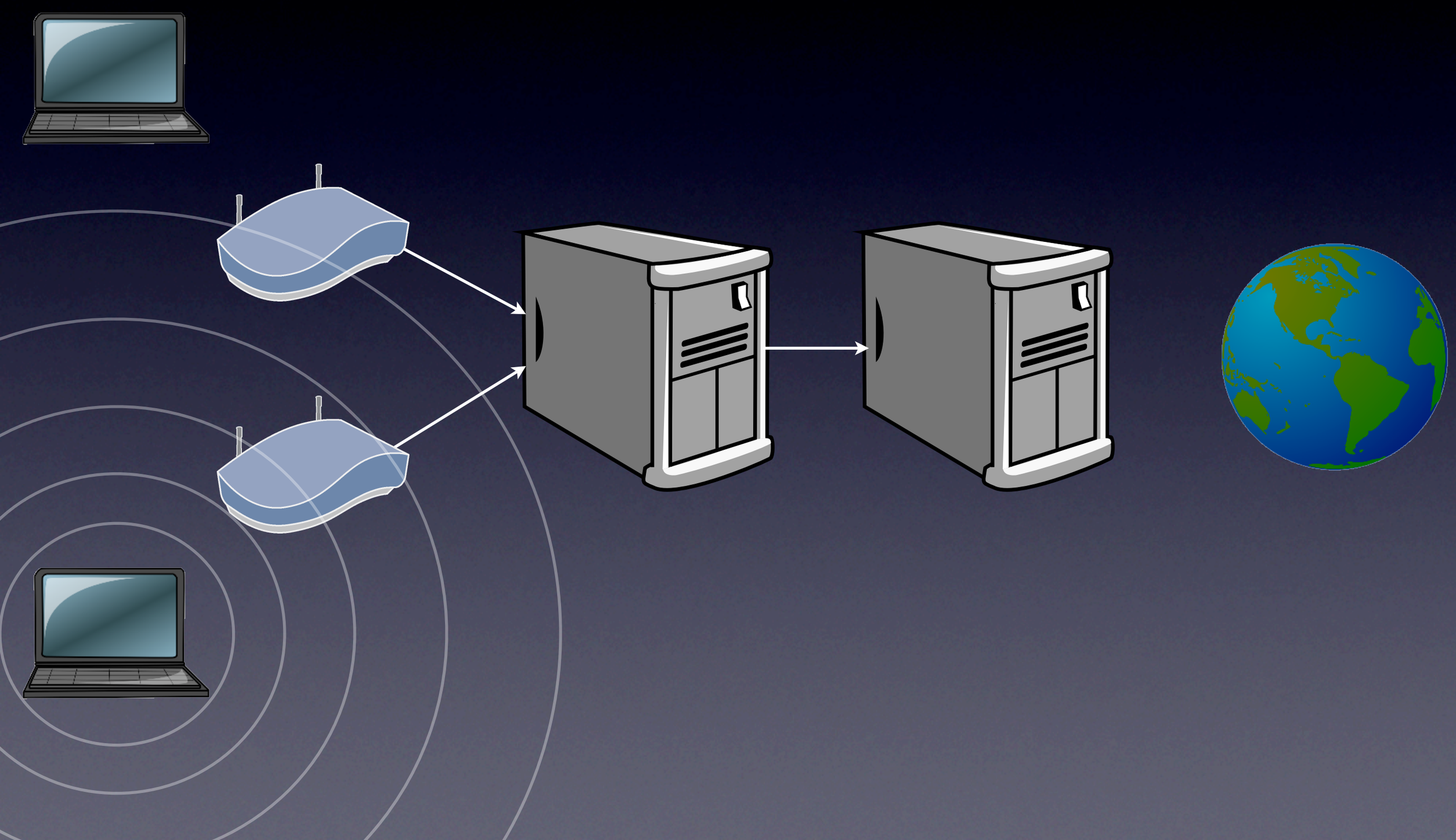
# Measure



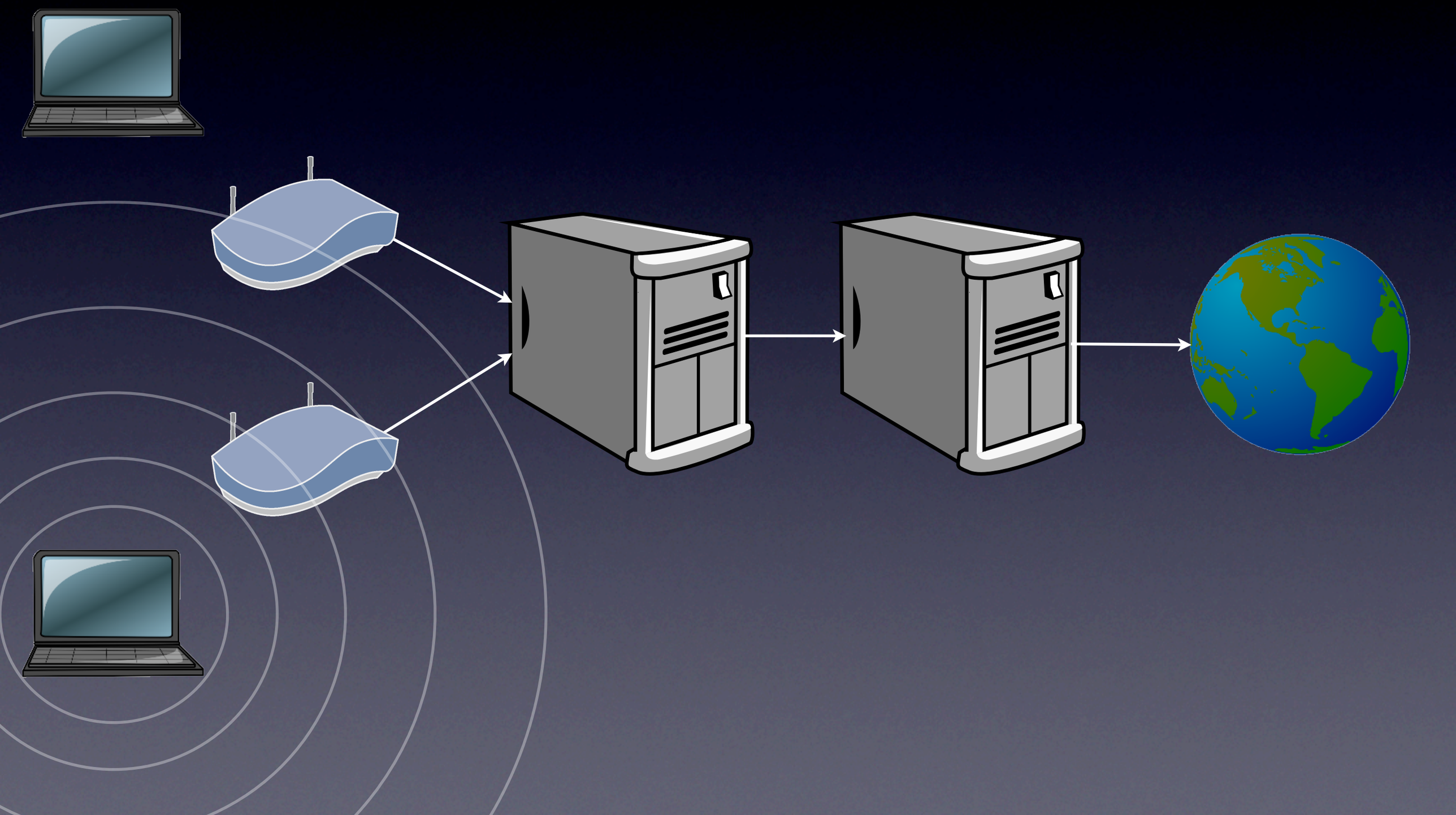
Measure Merge



Measure Merge Analyse



Measure Merge Analyse Publish

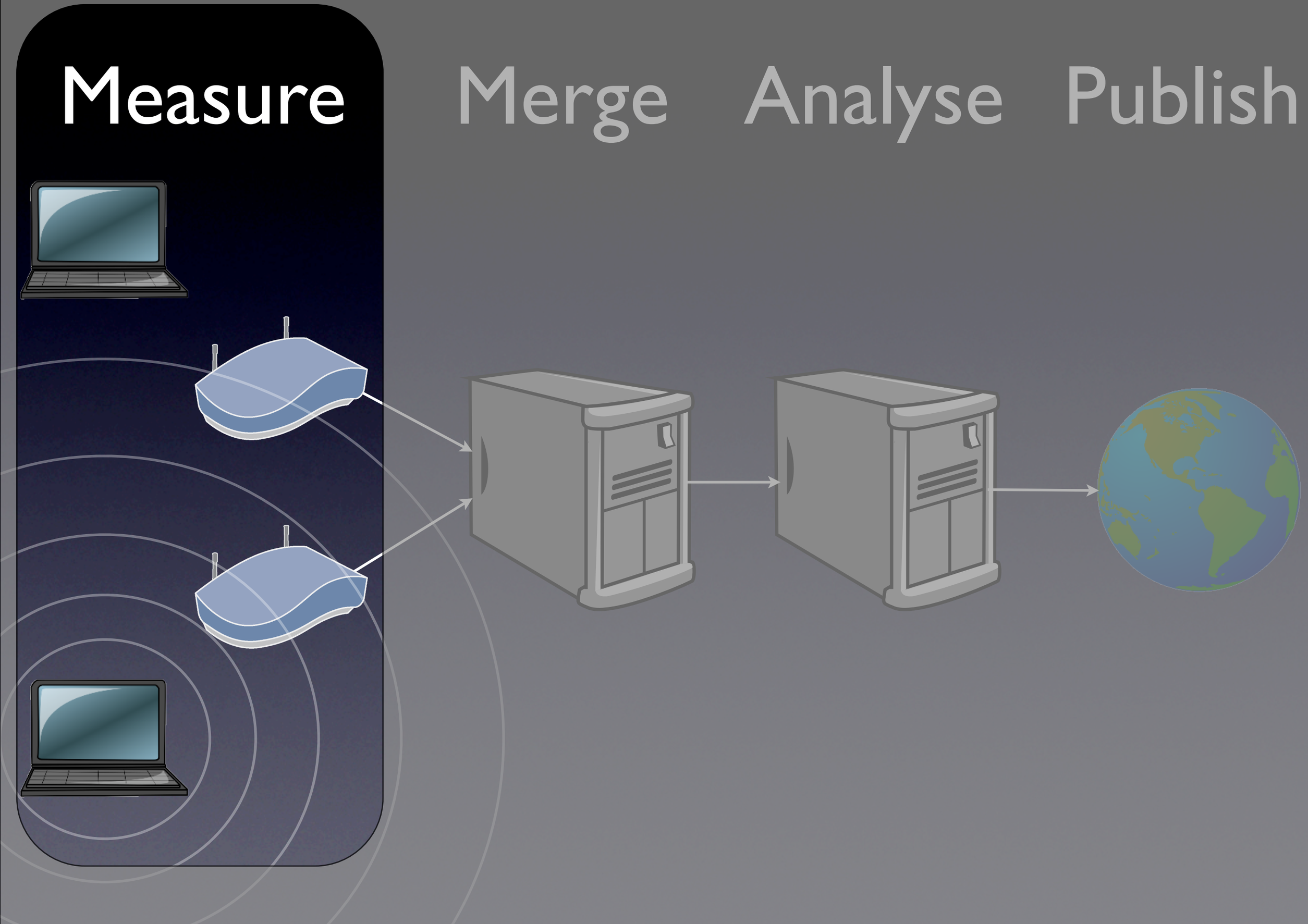


Measure

Merge

Analyse

Publish

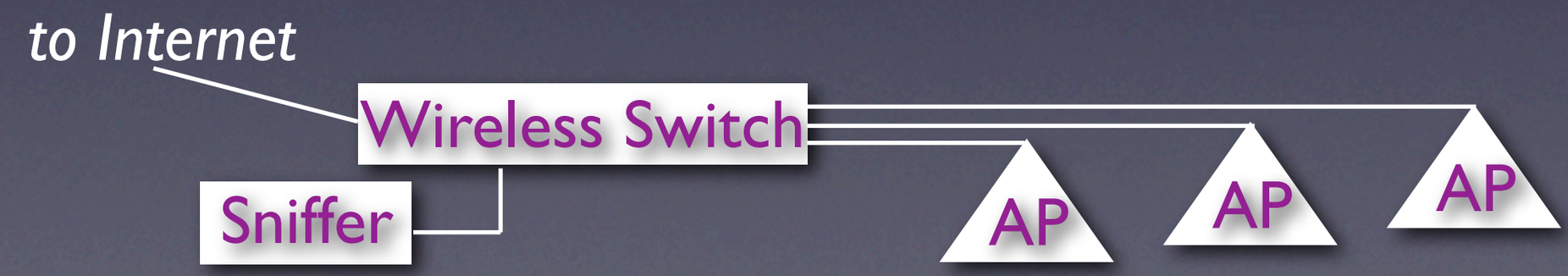


# Wireless measurement is hard

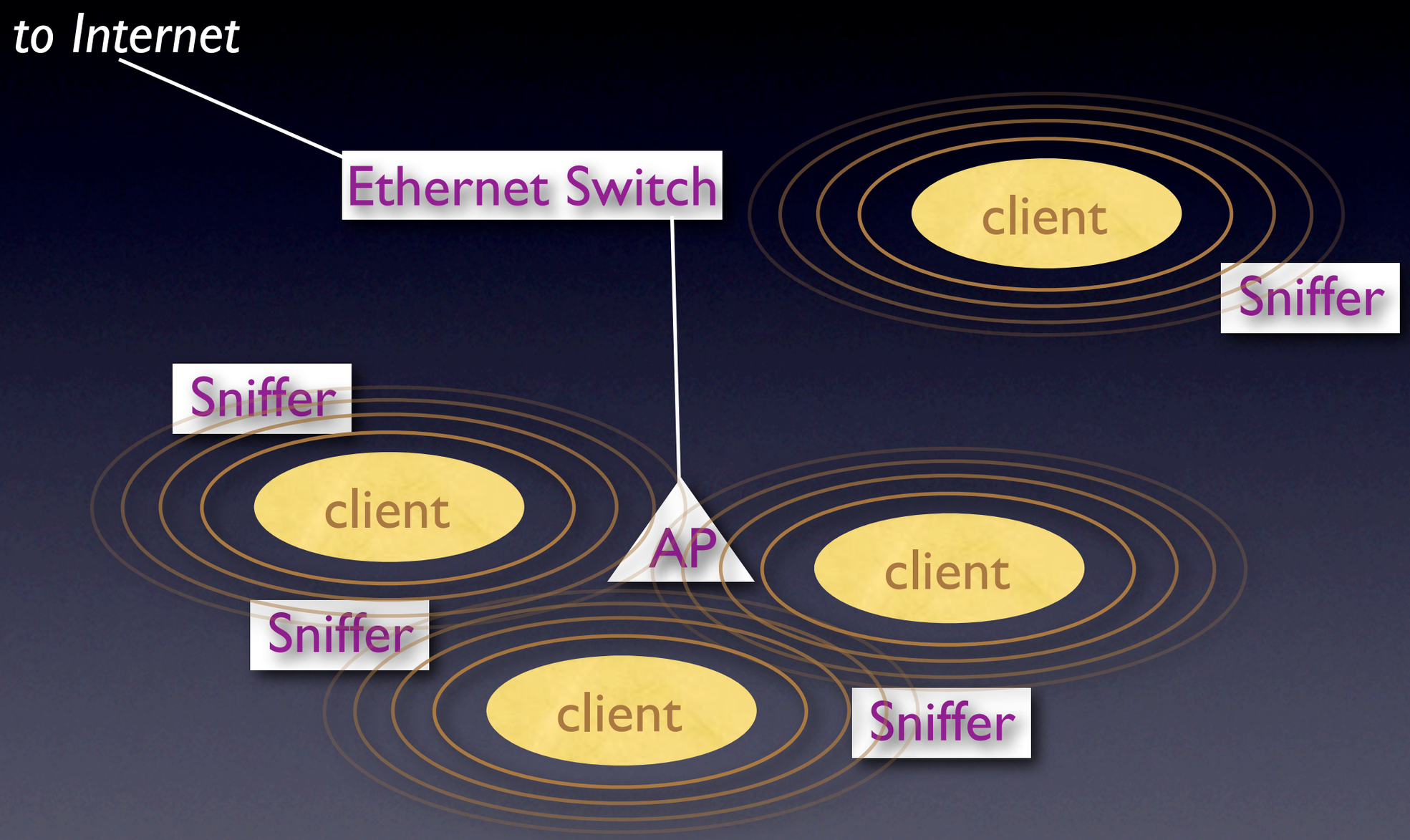
- hearing all wireless frames is hard
  - radio range/interference, loss, reflections
  - data flows can be huge (54Mbps and rising)
  - lack of portable tools, standard data formats, open driver interfaces
  - modern APs can dynamically alter power levels and channel assignment
  - we need to sniff the air...

# Why sniff the air?

- Why not measure wired side of APs?
  - only observe bridged traffic
  - no management traffic, errors, attacks, etc
  - only observe known networks
  - only observe 802.11



# “sniffing the air”




<dfk@cs.dartmouth.edu>

June 1, 2006

# What's wrong with existing tools?

- Don't hear everything
  - war-driving only needs to hear beacons
- Aren't tested
  - often just use data frames to test performance
- Don't scale
  - many commercial products designed with multiple sensors, but only few packet captures at a time

# Case study: MobiSys 2005

- deployed 3 multi-radio wireless sniffers
  - no wired network connectivity: hard to reconfigure
- Problems:
  - AP channels were reconfigured multiple times during conference
  - sniffers were not optimally positioned
- Result: lossy/corrupted data set

# How could we do better?

- sniffers could be aware of changes in network configuration
  - e.g., AP changes channel → sniffer changes channel
- sniffers could be reconfigured remotely
  - without wired connectivity
- sniffers could be optimally positioned
  - or leverage existence of other sniffers to aid capture

# Problem: channel sampling

- 14 channels (802.11b/g), 20+ channels (802.11a)
- Each sniffer can only listen to one channel
  - But interesting traffic may be on other channels
- Channel-hop strategically
  - spend more time on "interesting" channels
  - e.g., highest frames/bytes/ESSIDs/BSSIDs/STAs/IBSSes
  - e.g., track a particular ESSID/BSSID/STA
  - e.g., largest change in a particular metric
  - e.g., non-802.11 signals (Bluetooth/microwave/etc.)

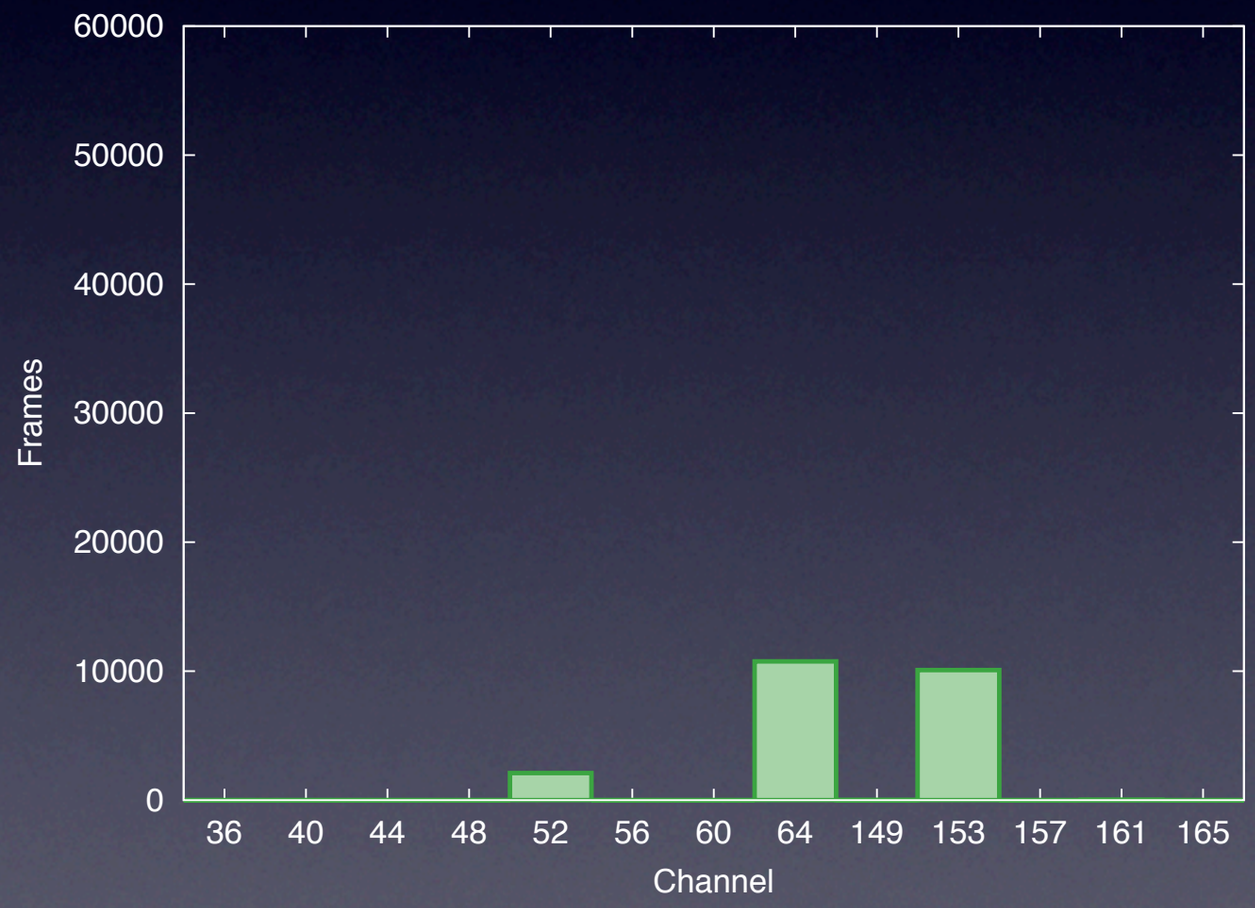
# e.g., Equal/Time


# e.g., Proportional/FrameCount

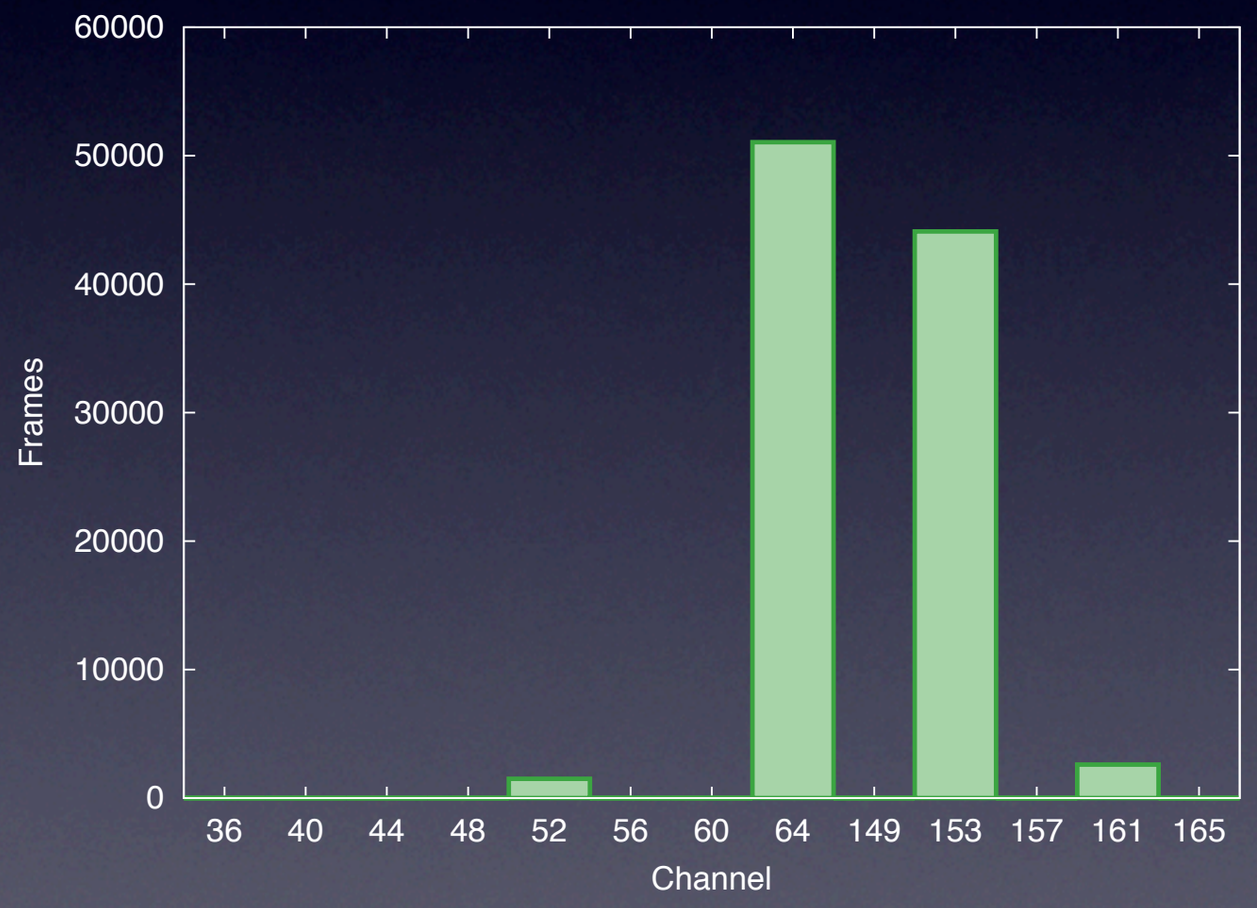

<dfk@cs.dartmouth.edu>

June 1, 2006

# Hear more on relevant channels



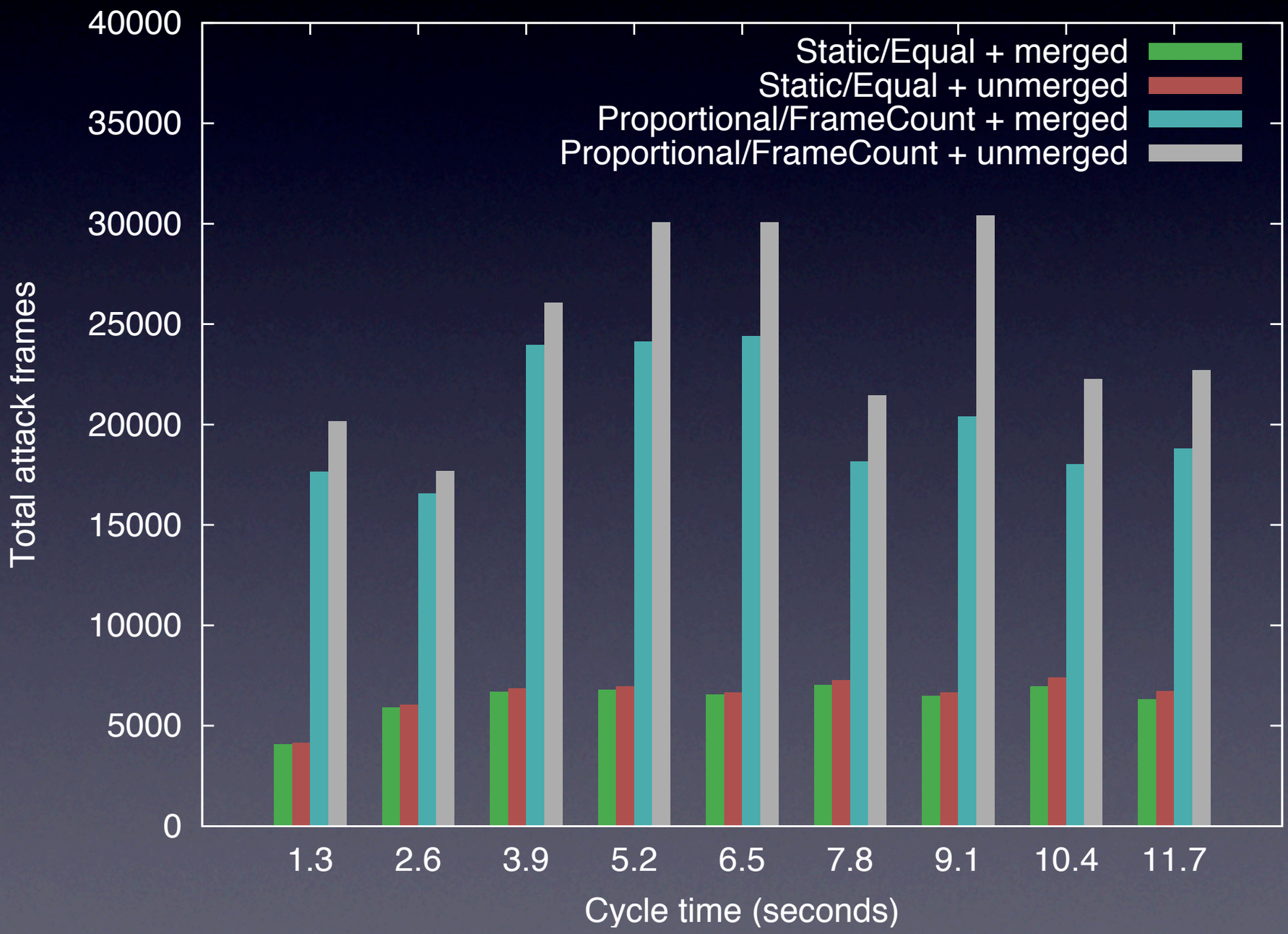

equal




proportional

# Hear relevant frames

## Sampling captures more attack frames

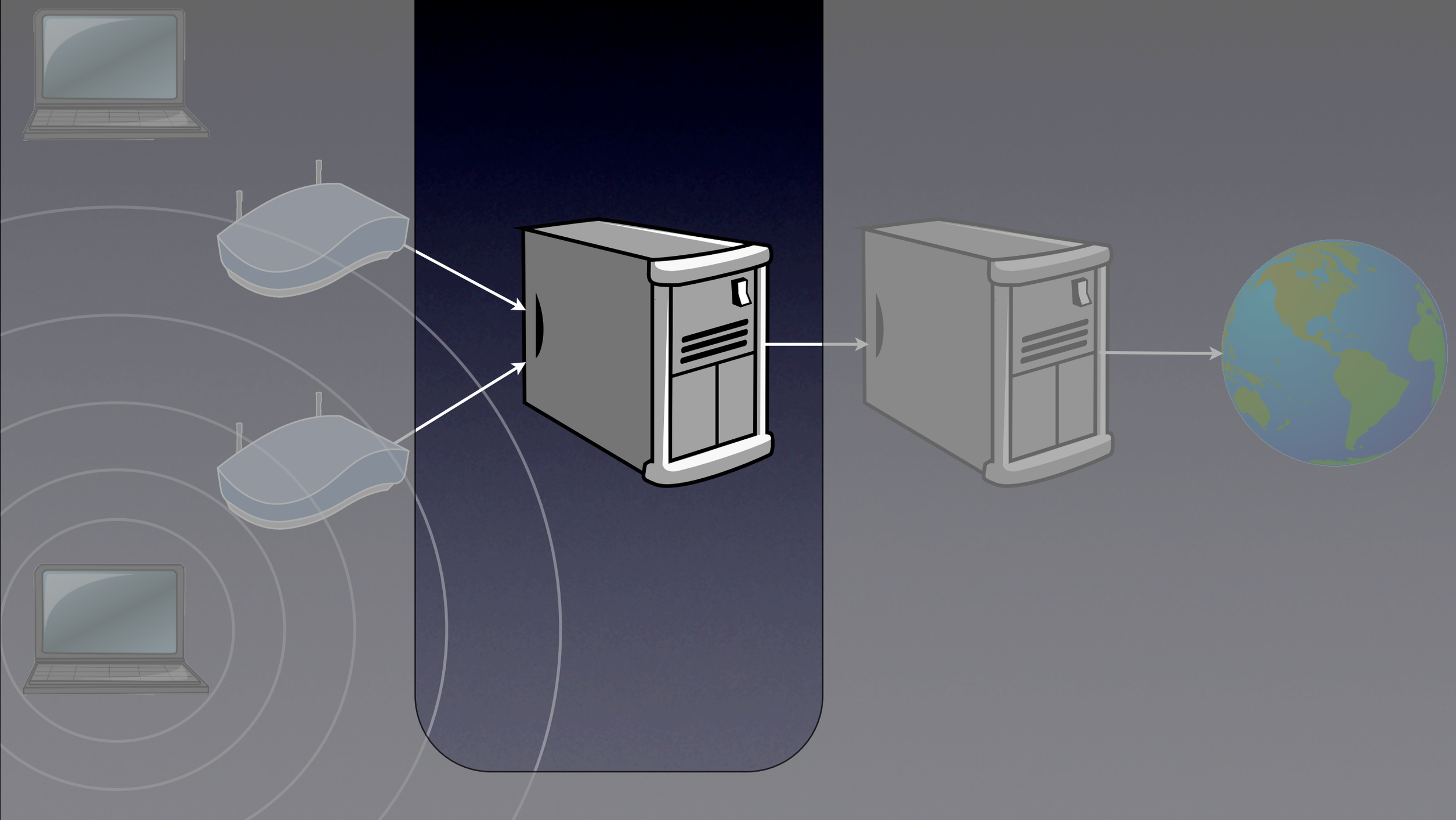


Measure

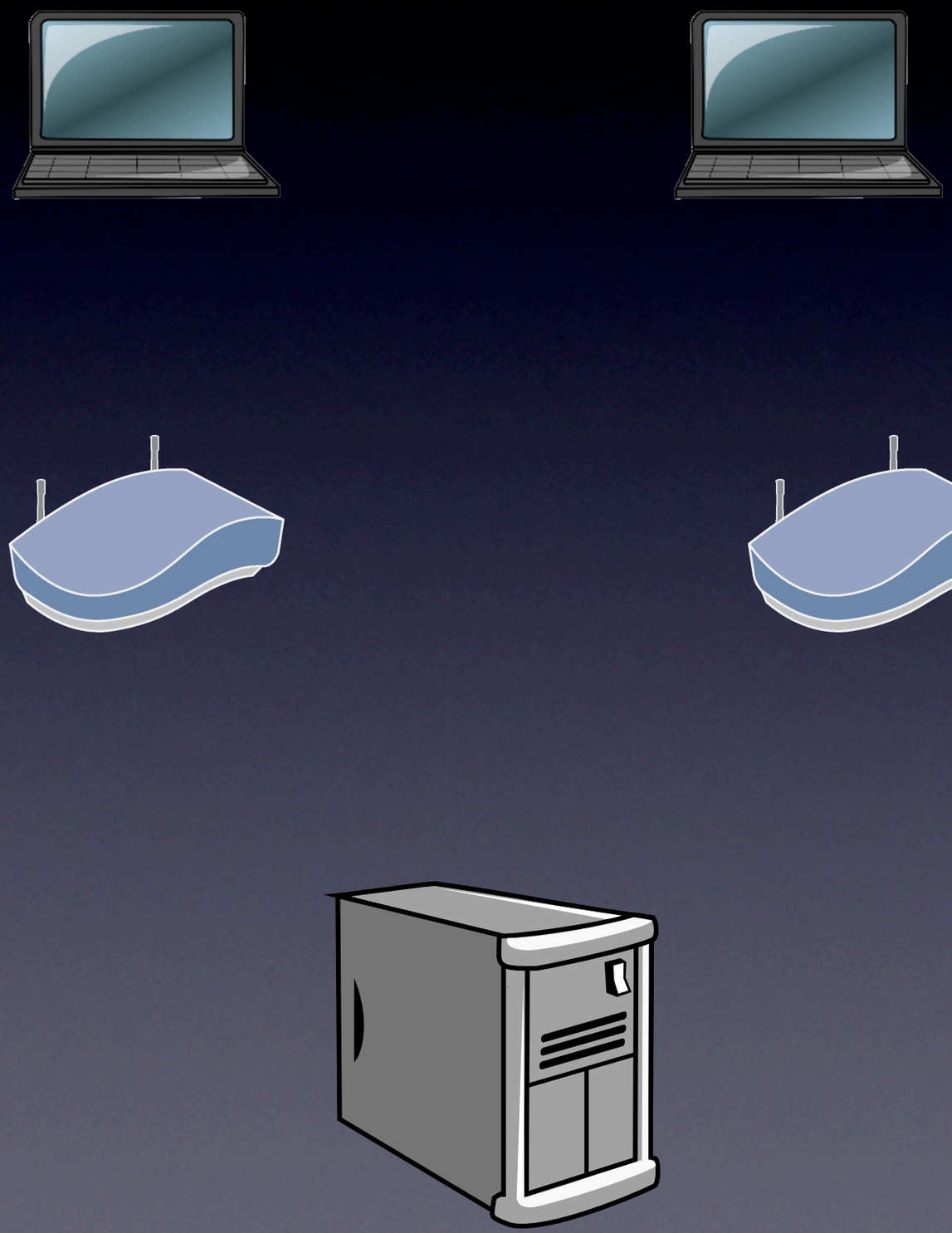
Merge

Analyse

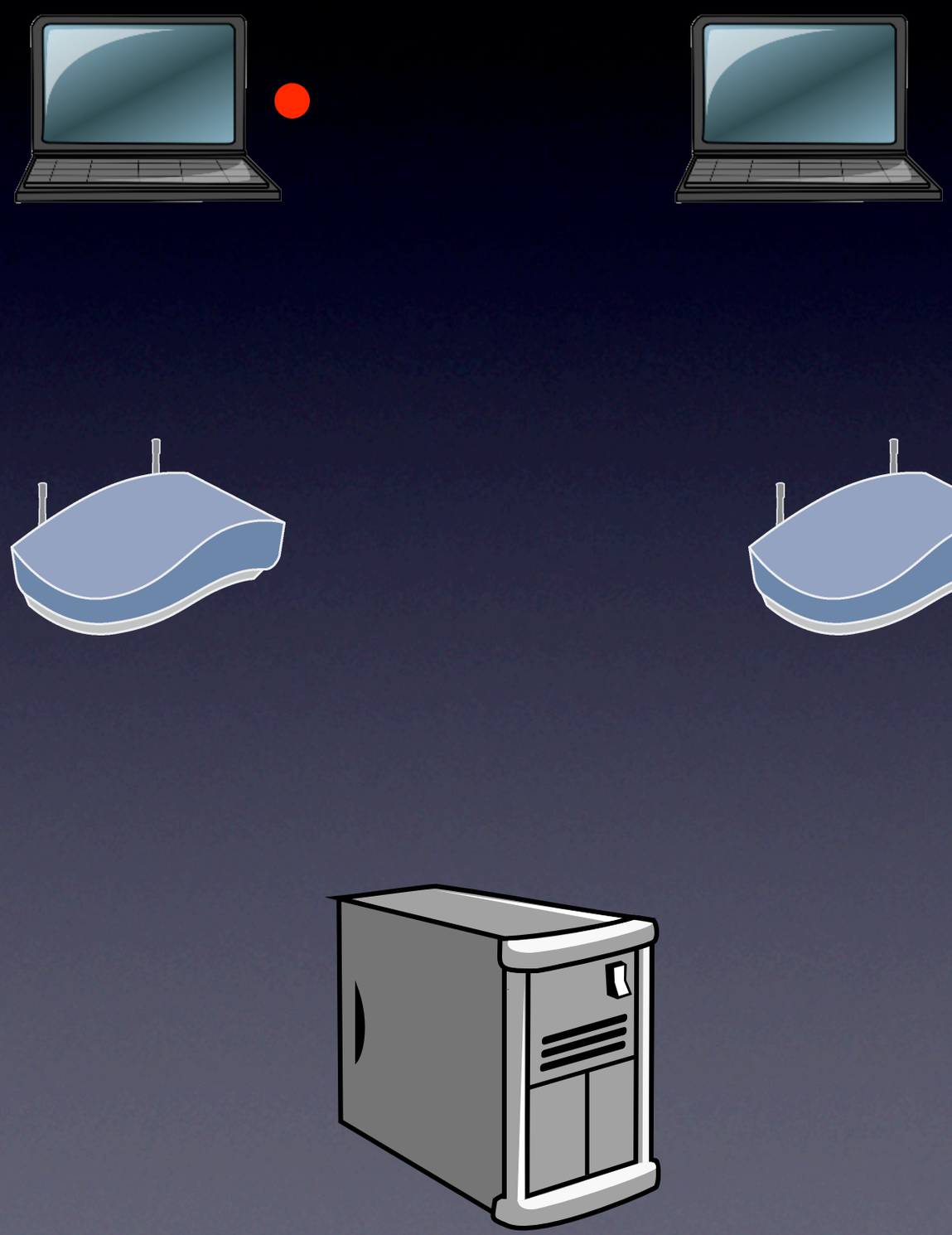
Publish



# Multiple sniffers, multiple captures

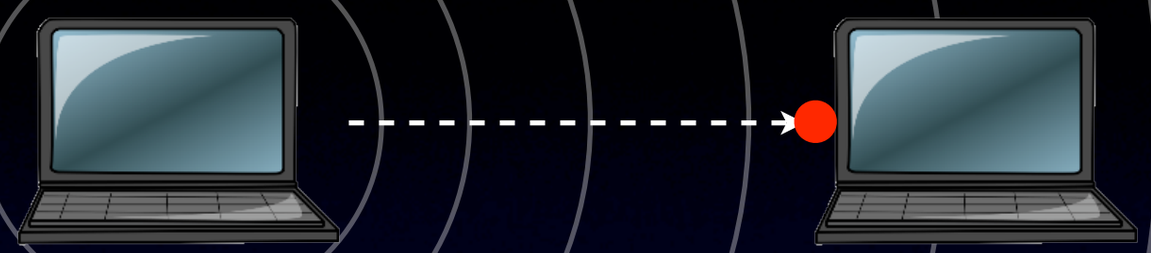


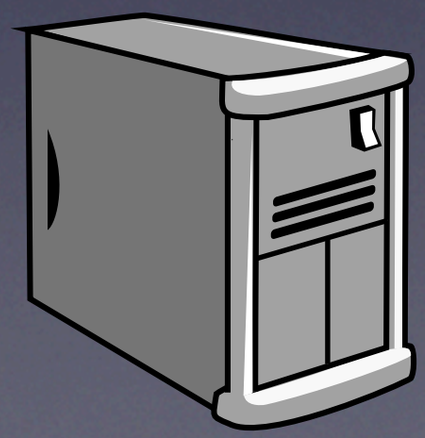
# Multiple sniffers, multiple captures



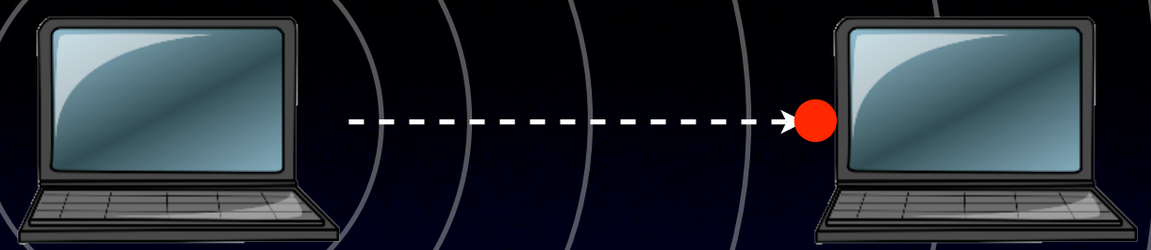
# Multiple sniffers, multiple captures

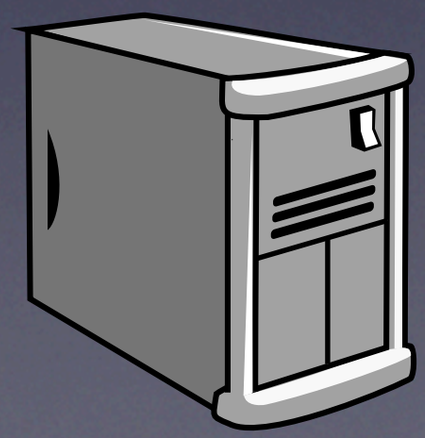
# Multiple sniffers, multiple captures



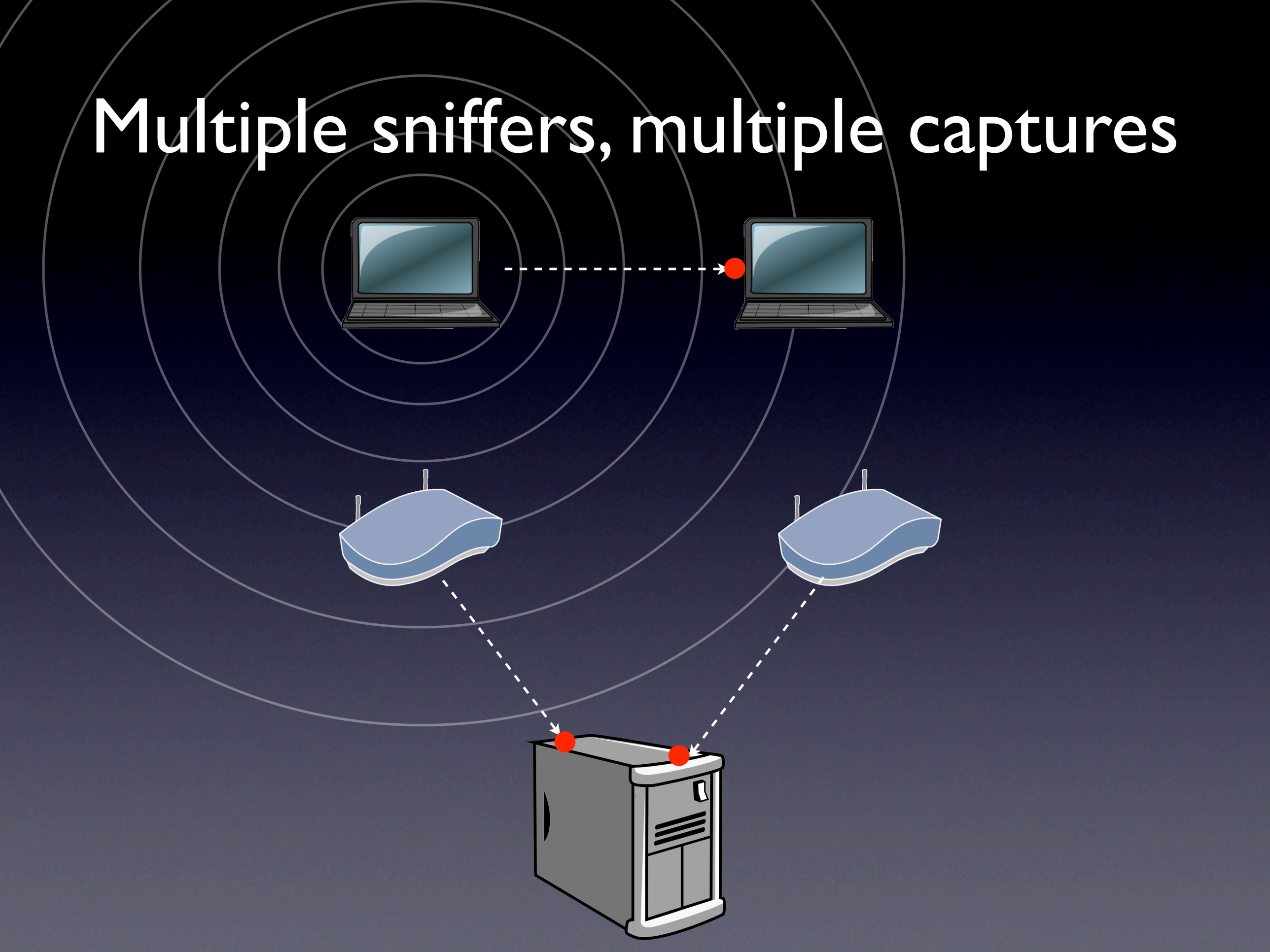


# Multiple sniffers, multiple captures

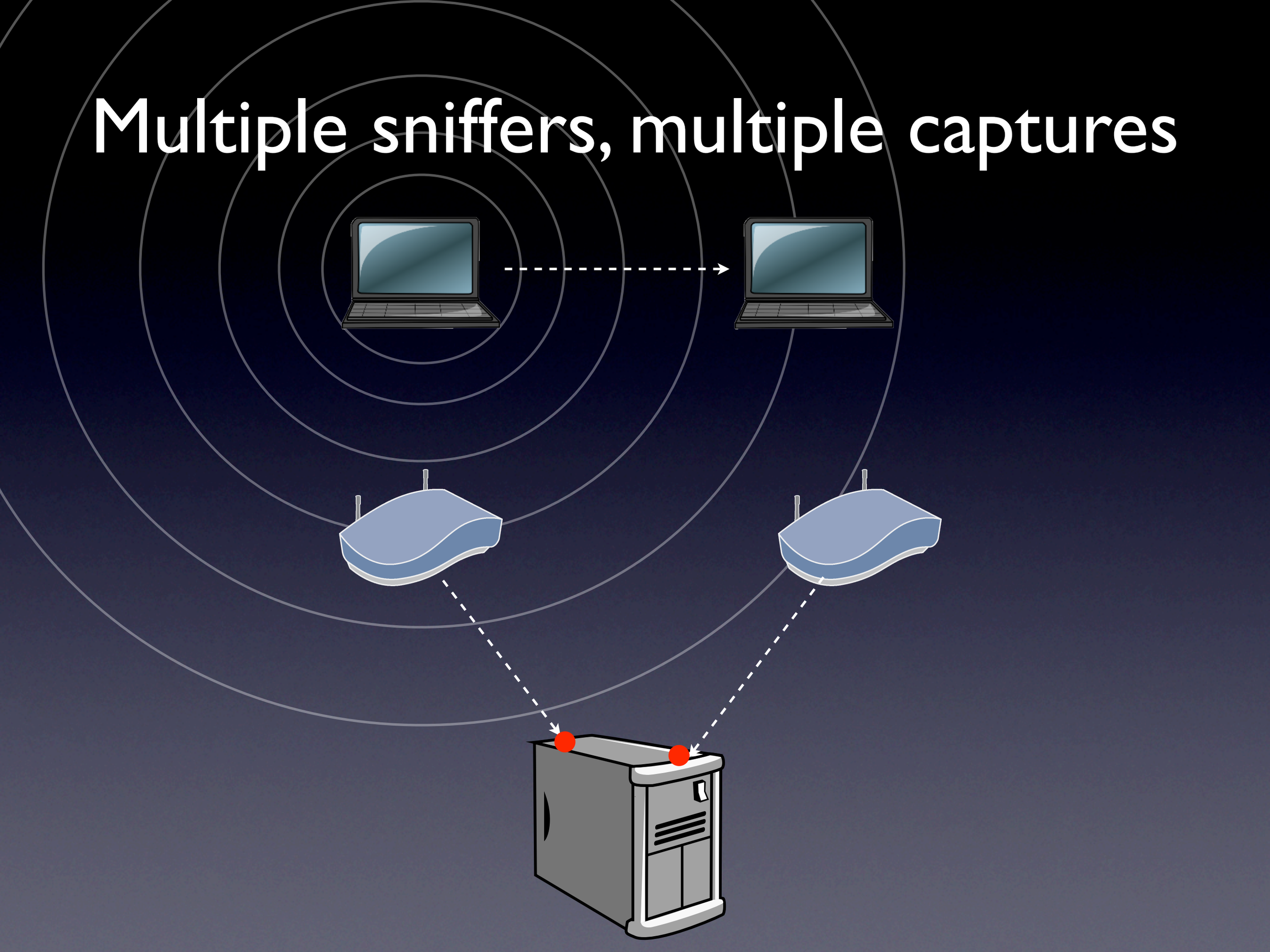




# Multiple sniffers, multiple captures



# Multiple sniffers, multiple captures

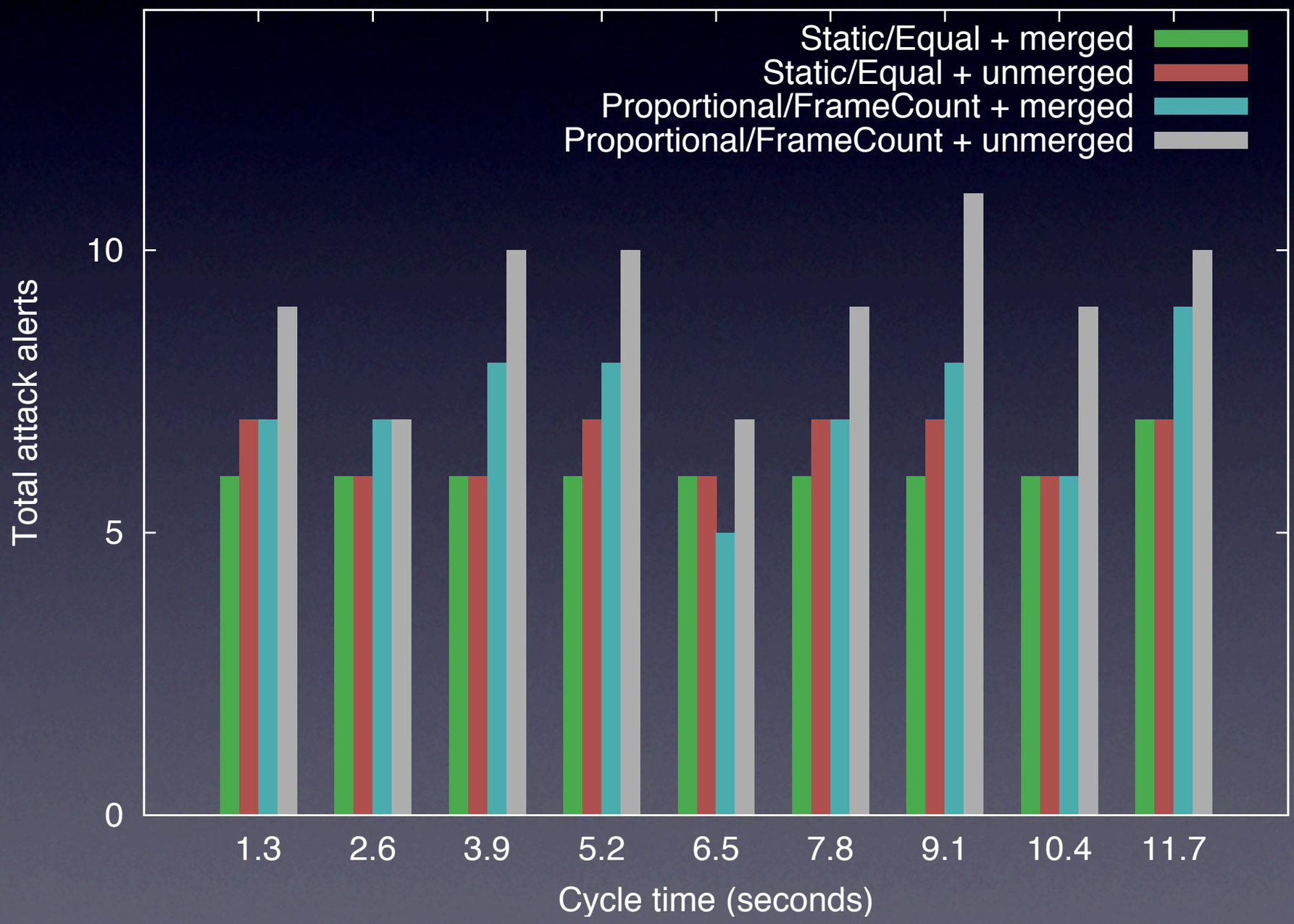


# Implementation

- Use beacon frames to synchronise clocks
  - this is difficult!
  - use TSF Timer as well as system clock
- Use FCS as keys in hash table
  - is this sufficient?
- Need to keep track of retransmits, reordering

# Merging gives more realistic view

## Fewer false alarms with merging

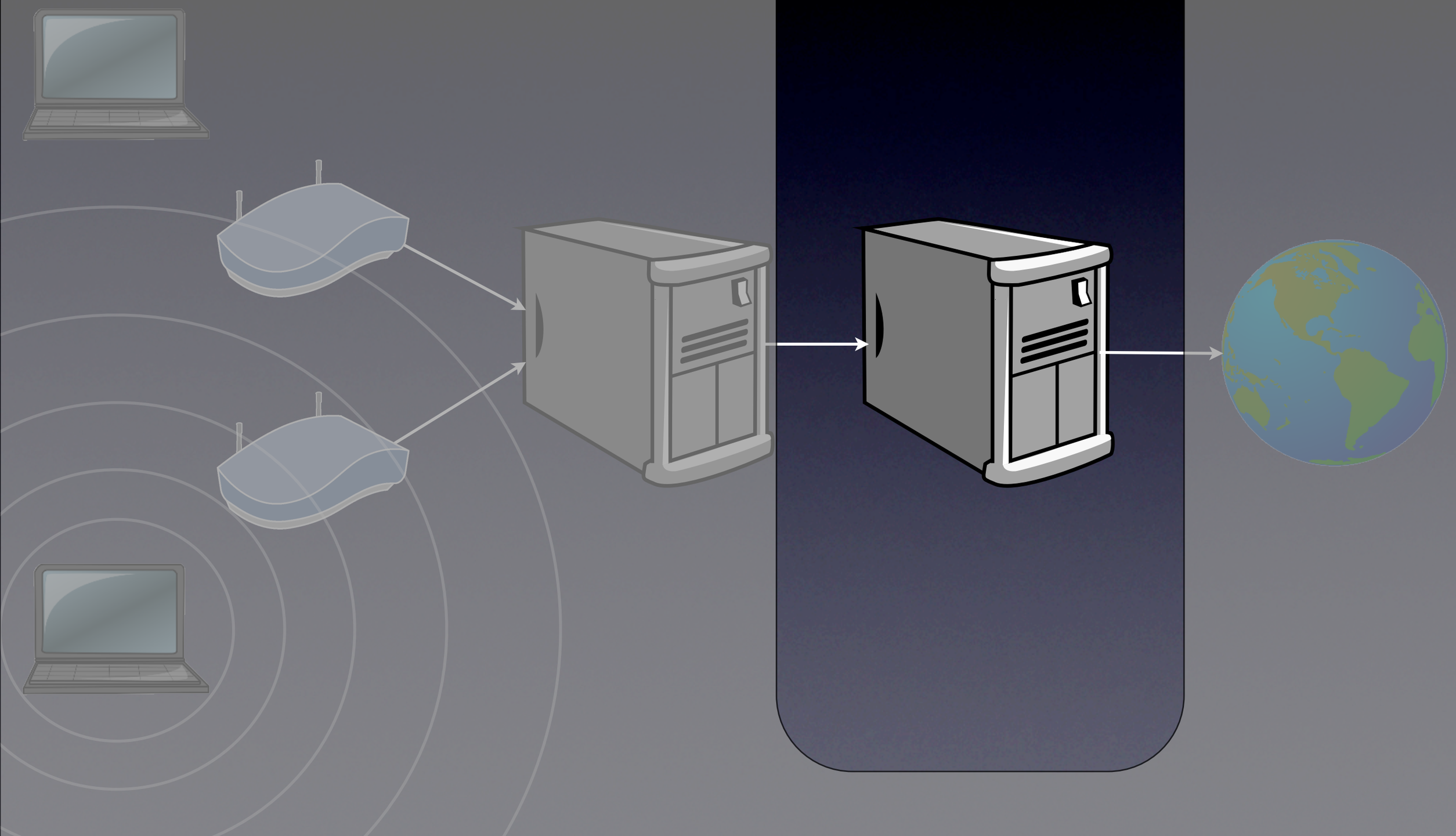


Measure

Merge

Analyse

Publish

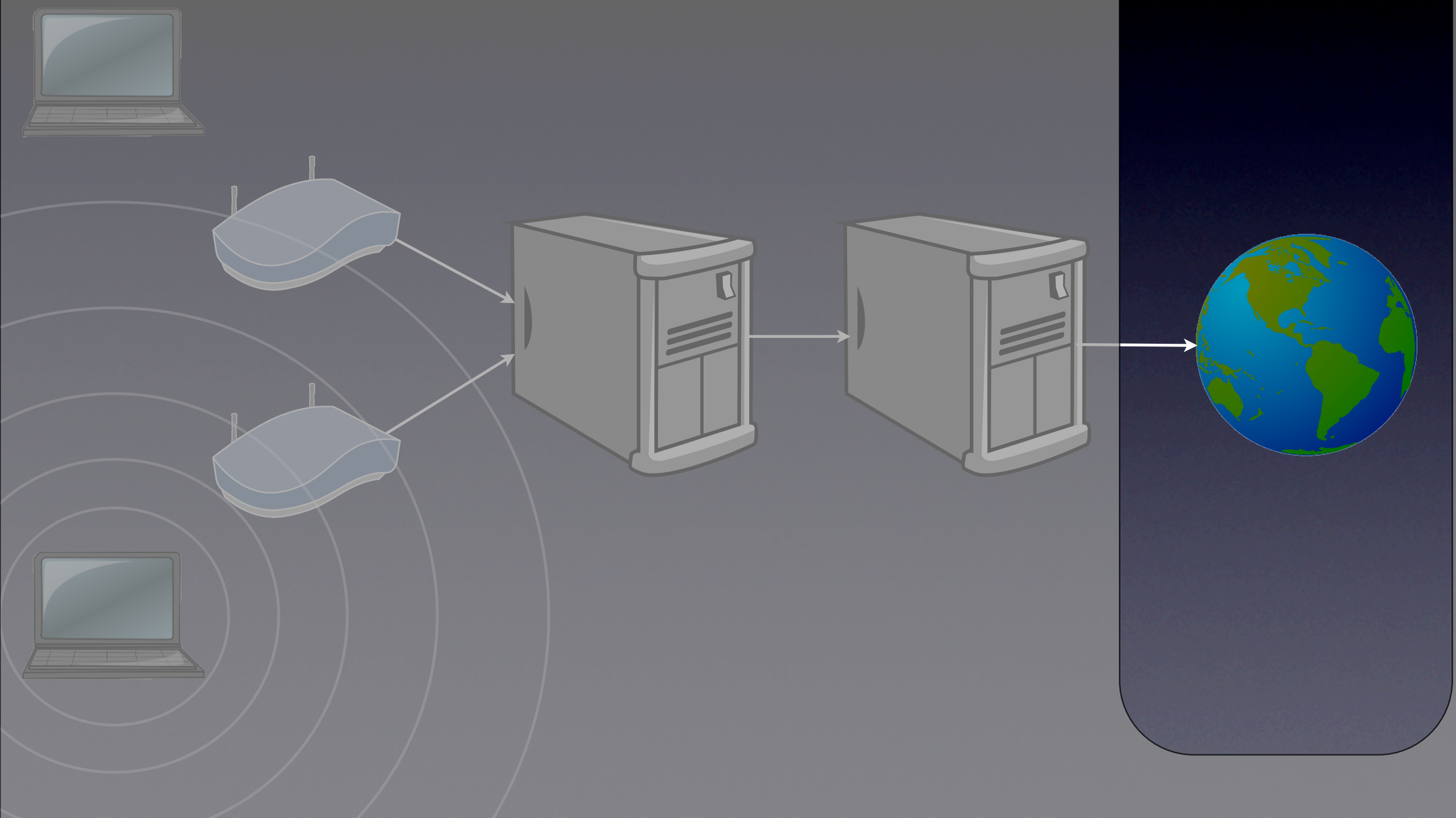


# Problem: measurement loss

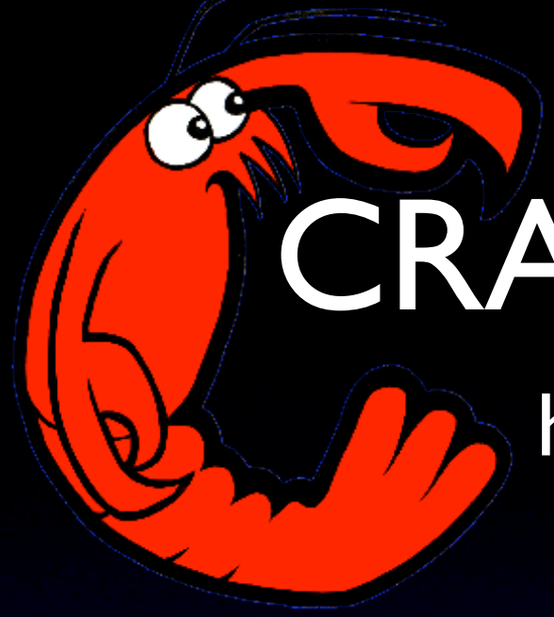
How do we know if we see a true picture of the air?

- how to verify that measurement is working?
  - e.g., what if sniffer is badly positioned?
- parse sniffer output to look for loss
  - recreate 802.11 FSM
    - e.g., look for DATA-ACK, RTS-CTS, etc
  - come up with ‘single number’ ‘executive summary’
    - i.e., move sniffer around until number is higher

Measure Merge Analyse Publish



# So, what do we do with all this data we've collected?



# CRAWDAD

http://crawdad.cs.dartmouth.edu

- Community Resource for Archiving Wireless Data At Dartmouth
  - Provide data for researchers, and tools to make it easy to collect more data
- 257 registered users
  - approximately 119 universities, 26 companies
- 13 data sets, and more coming
  - infrastructure/MANET/VANET/Bluetooth/etc.

<dfk@cs.dartmouth.edu>

June 1, 2006

# Problem: sanitising wireless traces

- Need to remove identifiable information from traces before release
  - federal (IRB) requirements, privacy risks, etc.
- Is it possible to "anonymise?"
  - how much is enough?
- Our tools:
  - remove everything >L4
  - sanitise IP addresses (prefix-preserving IP anonymisation, Xu et. al., ICNP 2002)
  - sanitise 802.11 identifiers (MAC addresses, ESSIDs)

# Challenges

- How can we best leverage multiple sniffers?
- How can we correlate with other data sources:
  - syslog, snmp, RADIUS, call manager, user location
- How can we verify that we are measuring well?
- How do we extract realistic mobility models?
- How can we protect users' privacy?
- How do we relate MACs to "users"?
- How do we identify different device types?
- How can we share the captured data?

# Challenges

- How can we best leverage multiple sniffers?
- How can we correlate with other data sources:
  - syslog, snmp, RADIUS, call manager, user location
- How can we verify that we are measuring well?
- How do we extract realistic mobility models?
- How can we protect users' privacy?
- How do we relate MACs to "users"?
- How do we identify different device types?
- How can we share the captured data?