Daedalus/GloMop Architectural Walkthrough

Daedalus Retreat, Lake Tahoe

June 18-20th, 1996

Steve Gribble and Todd Hodes

{gribble,hodes}@cs.berkeley.edu

Outline

- Motivation and overview
- Core architecture components
- Core architecture walkthrough
  - connection, document retrieval, vertical handoff, document refinement
- Extended architecture components
- Extended architecture walkthrough
  - connection and authentication, vertical handoff, horizontal handoff, dynamic adaptation, load balancing, domain and network services
- Summary, Open Issues

Outline

- Motivation and overview
- Core architecture components
- Core architecture walkthrough
  - connection, document retrieval, vertical handoff, document refinement
- Extended architecture components
- Extended architecture walkthrough
  - connection and authentication, vertical handoff, horizontal handoff, dynamic adaptation, load balancing, domain and network services
- Summary, Open Issues

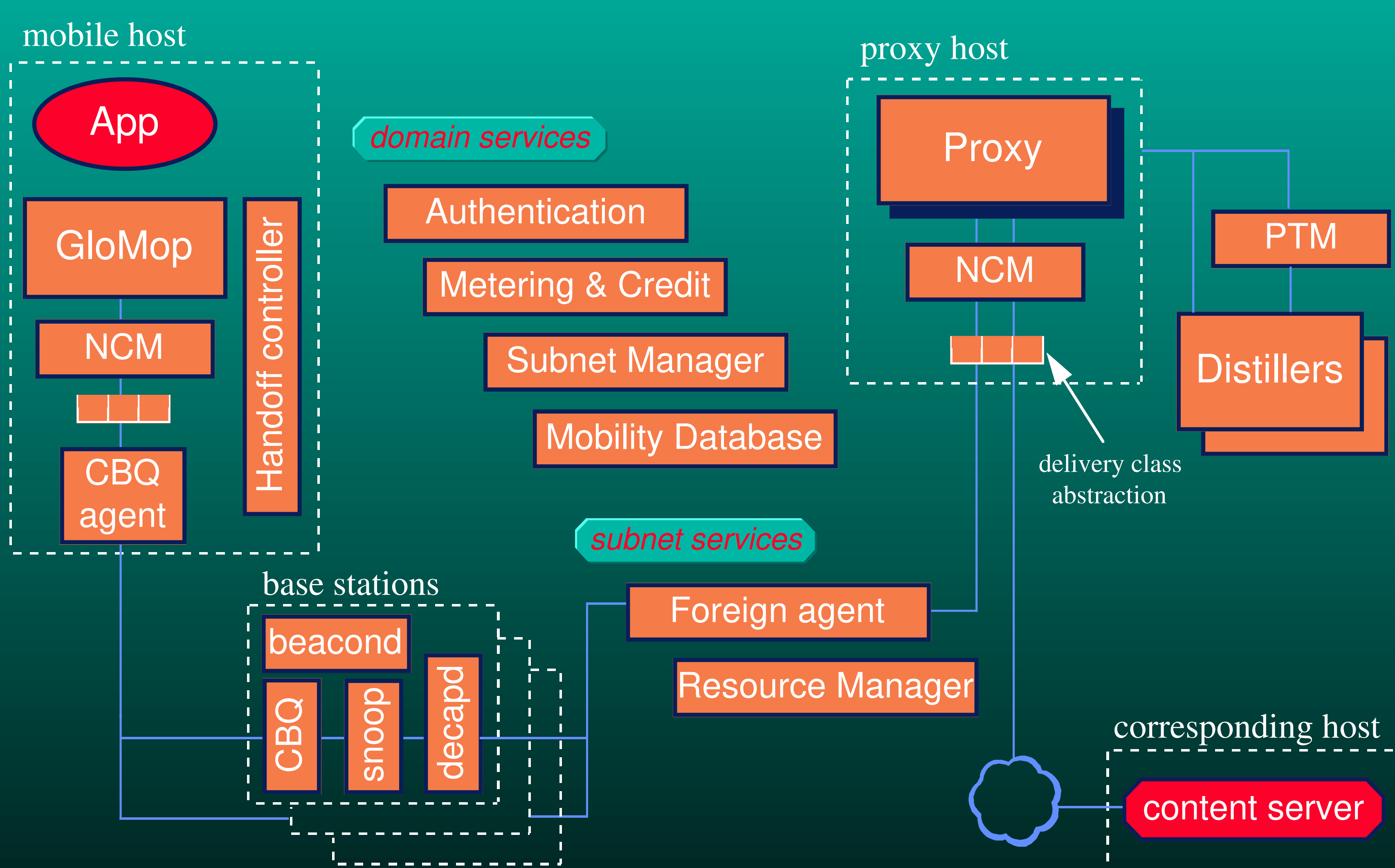
Motivation

- mobility-aware network services
  - seamless roaming (overlay IP)
  - efficiency of protocols and of handoff
- mobility-aware application services
  - dynamic adaptation to variability
  - enable a wide array of client devices
- mobility-aware domain services
  - resource discovery
  - load balancing
  - metering

Challenges

- cohabitate with existing infrastructure
  - Mobile IP, TCP, HTTP, Kerberos, ...
  - justify custom protocols
- a scalable architecture
  - network and proxies should support many clients
- integrable but independent components
  - pieces of the architecture should be able to function on their own
  - integration yields optimization and new capability

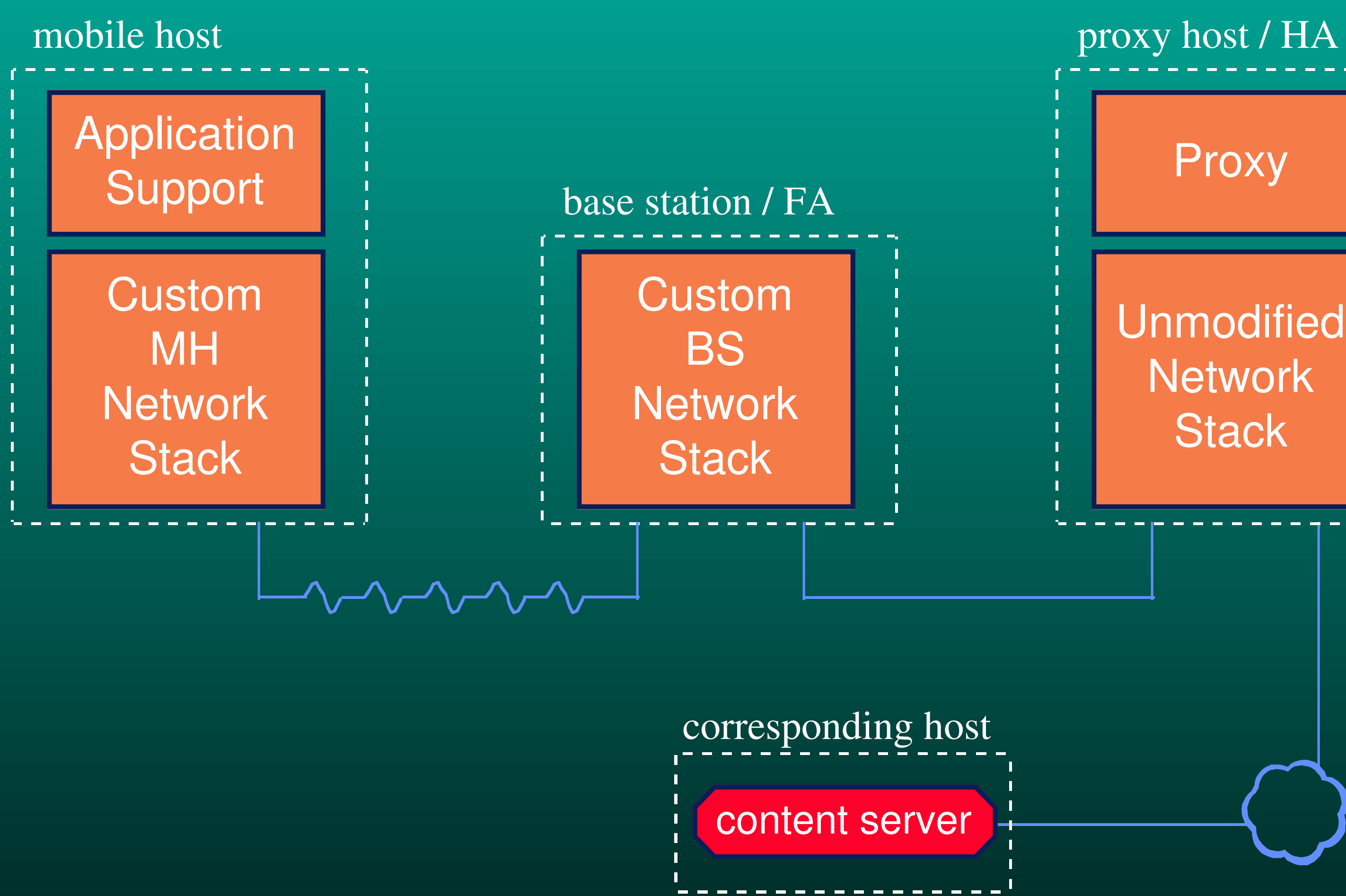
The Daedalus/GloMop Architecture



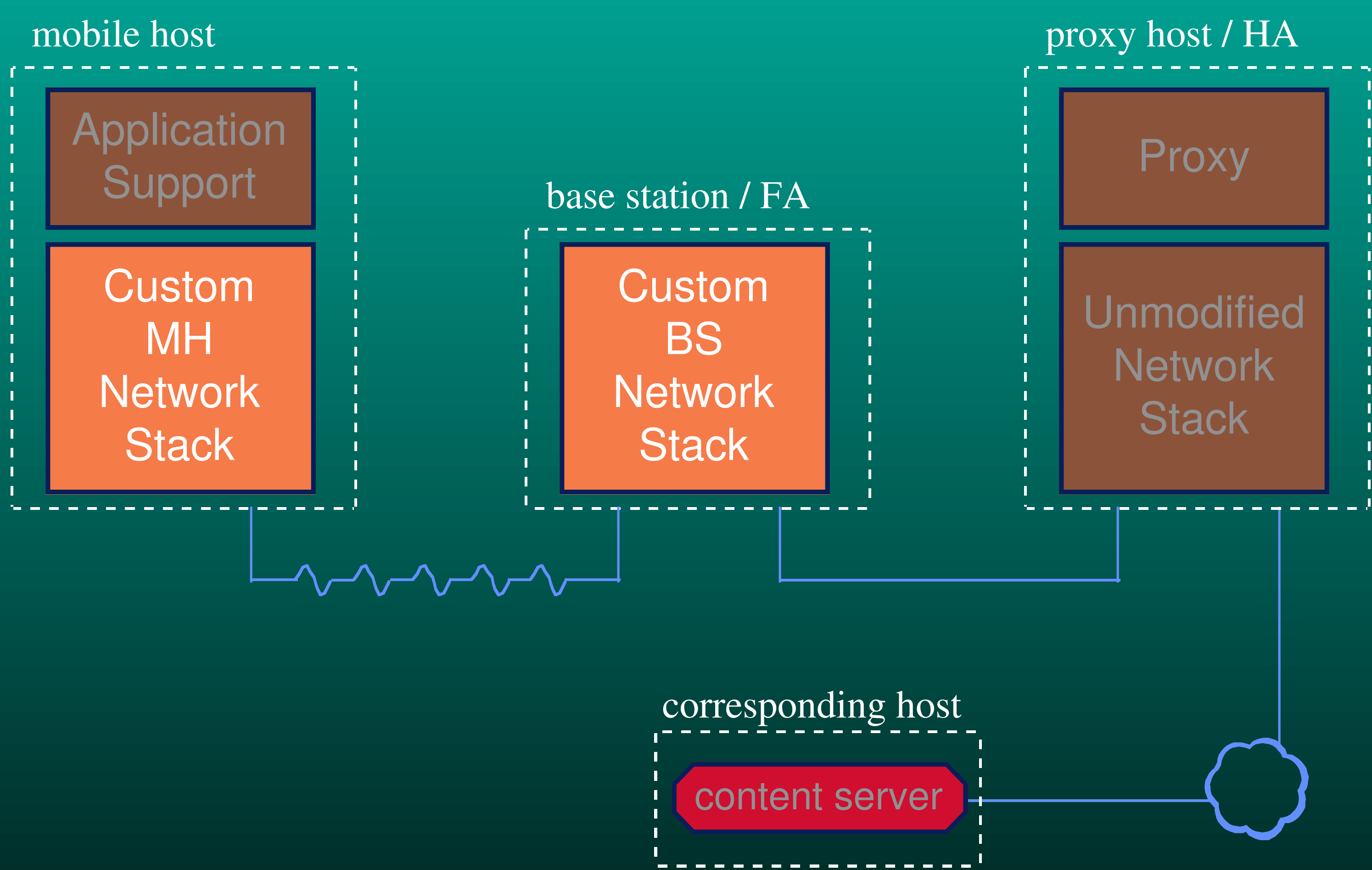
Outline

- Motivation and overview
- Core architecture components
- Core architecture walkthrough
  - connection, document retrieval, vertical handoff, document refinement
- Extended architecture components
- Extended architecture walkthrough
  - connection and authentication, vertical handoff, horizontal handoff, dynamic adaptation, load balancing, domain and network services
- Summary, Open Issues

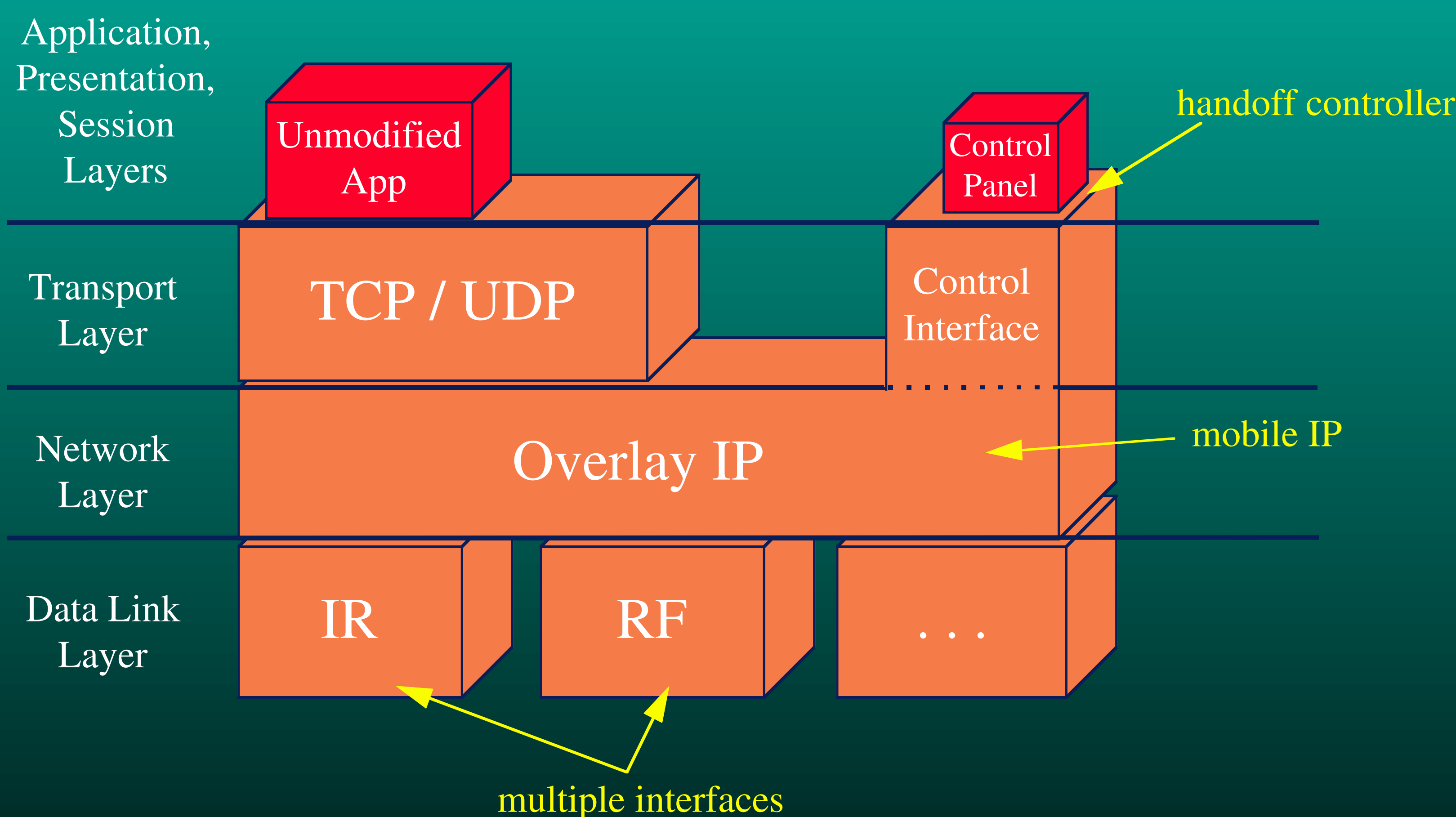
Core Architecture Components



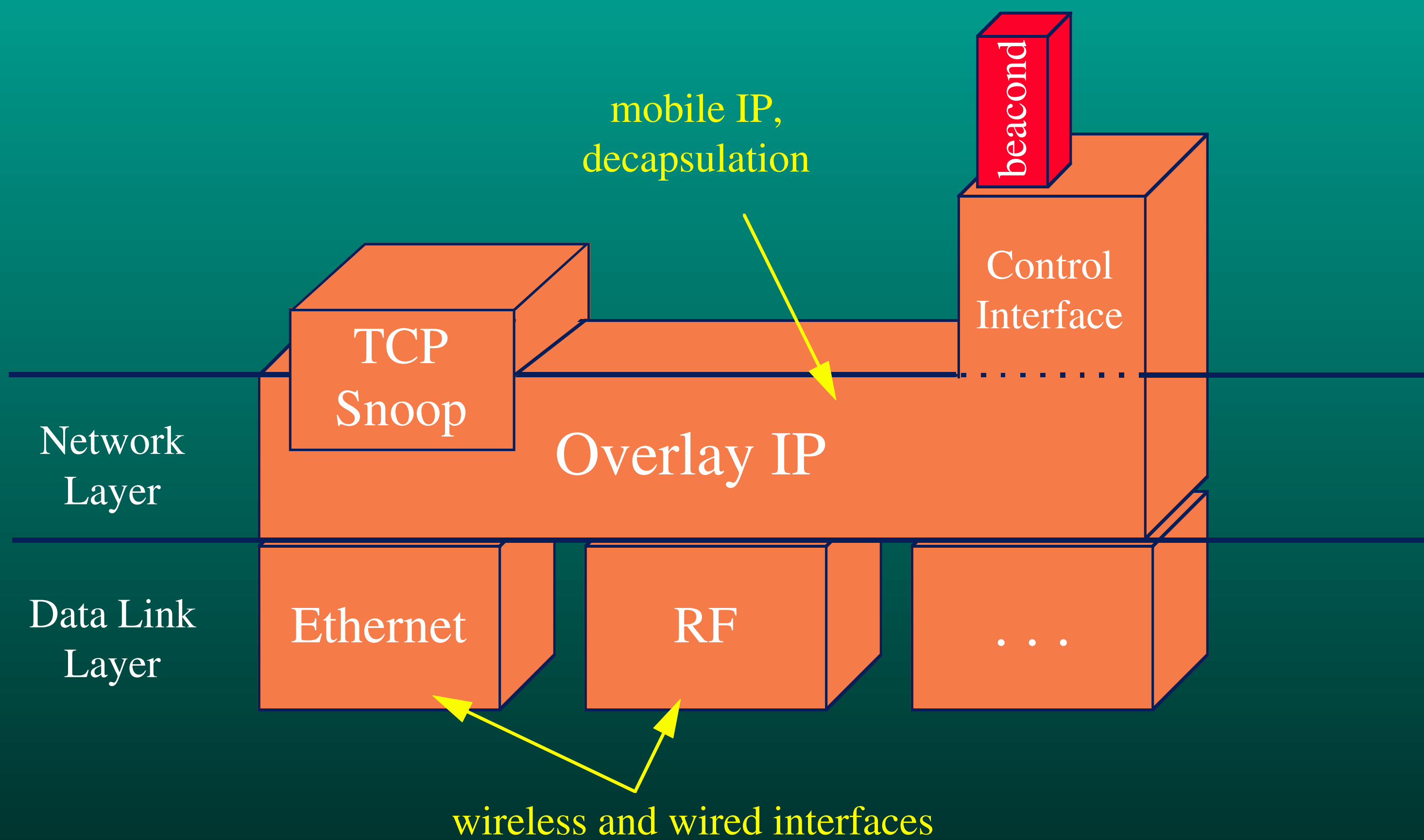
Core Architecture Components



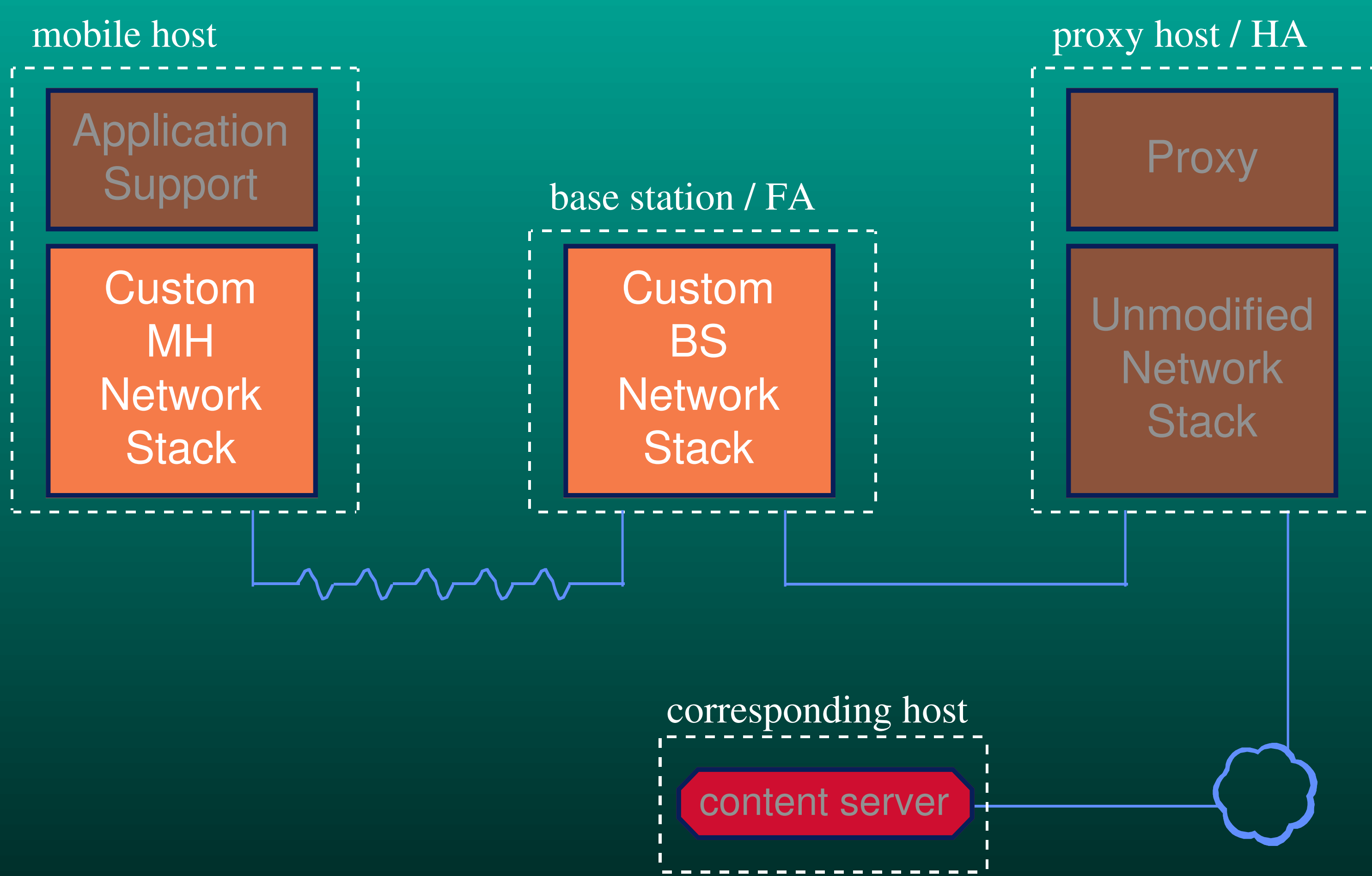
Core Daedalus Mobile Host Components



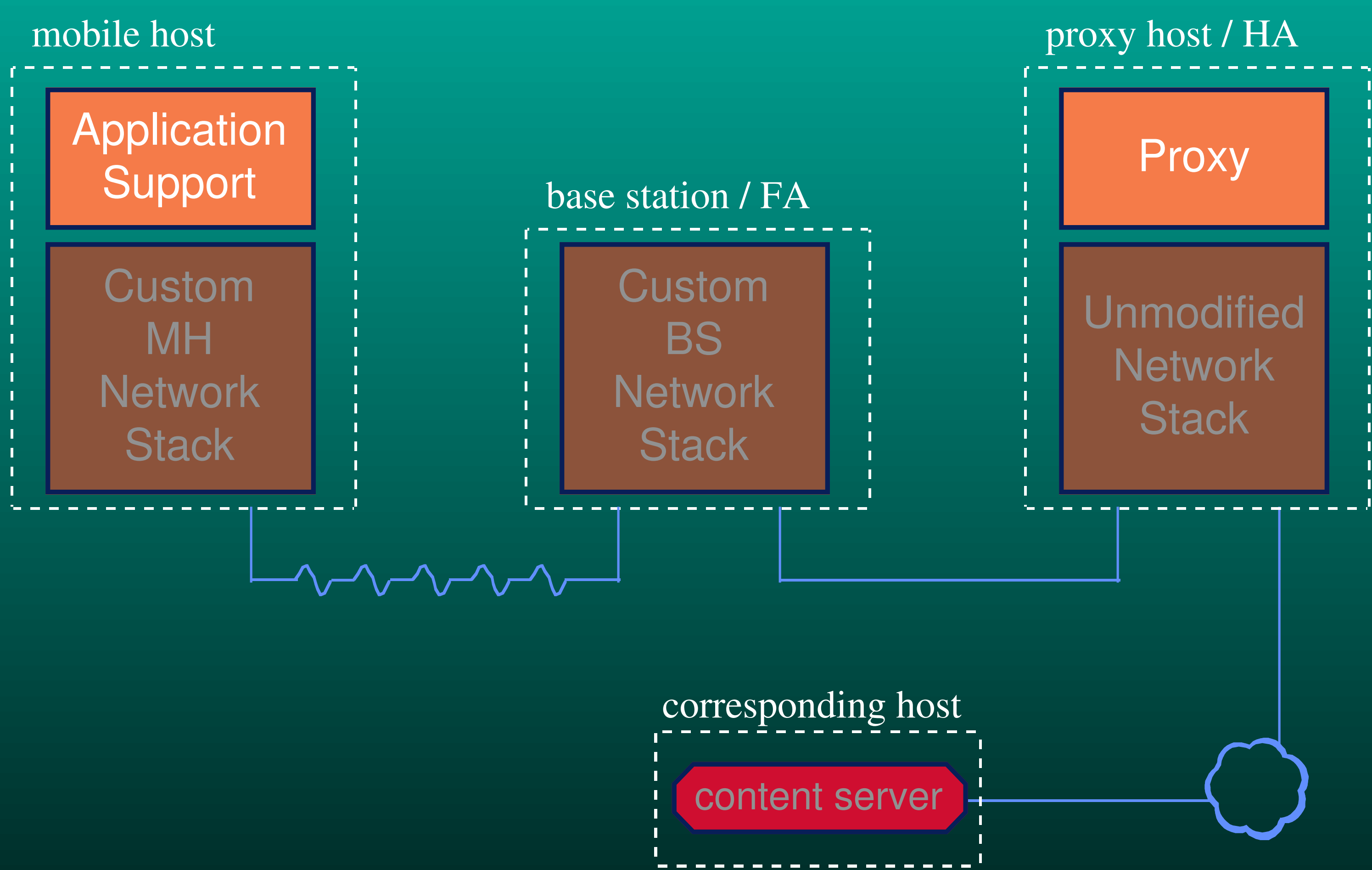
Core Daedalus Base Station Components



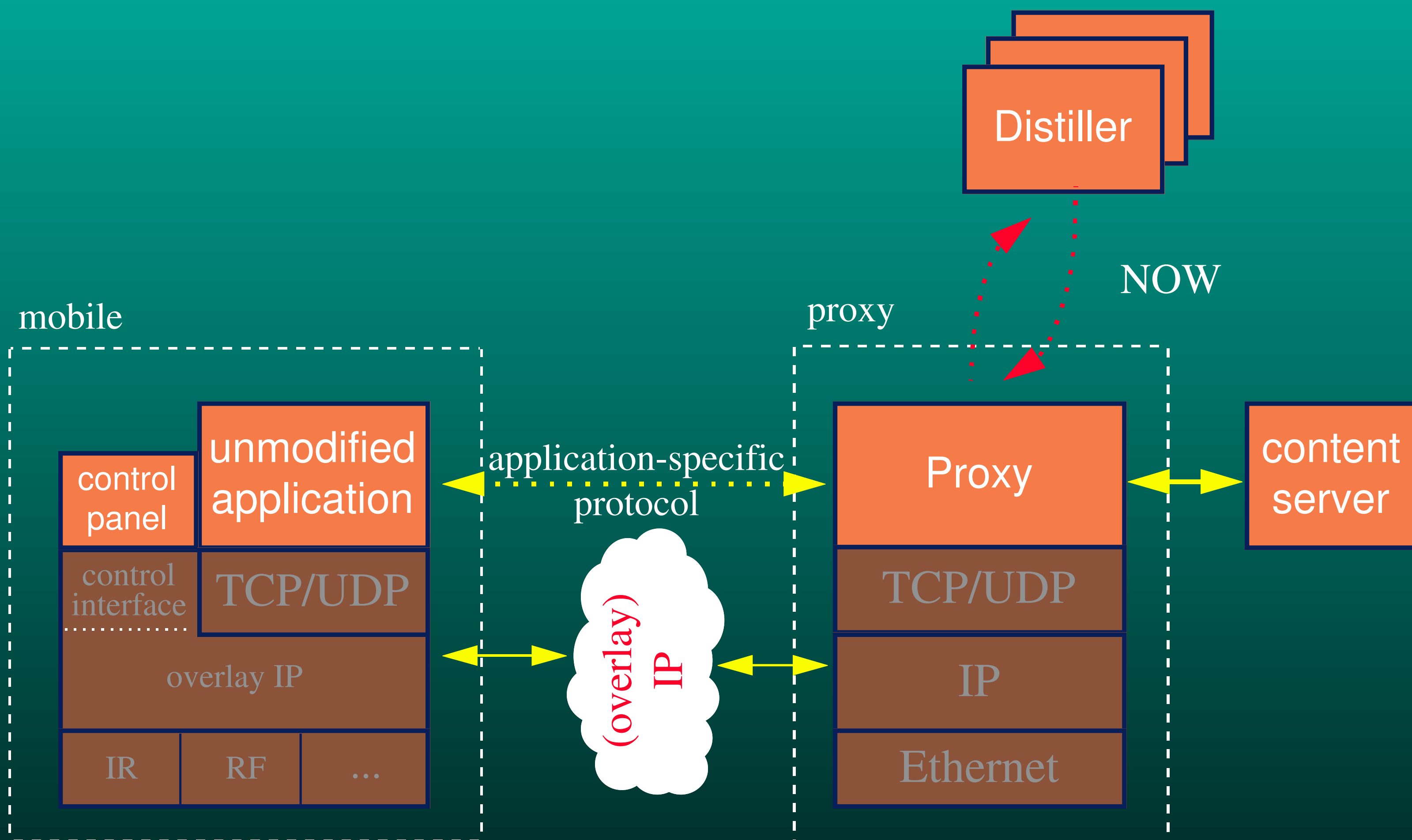
Core Architecture Components



Core Architecture Components



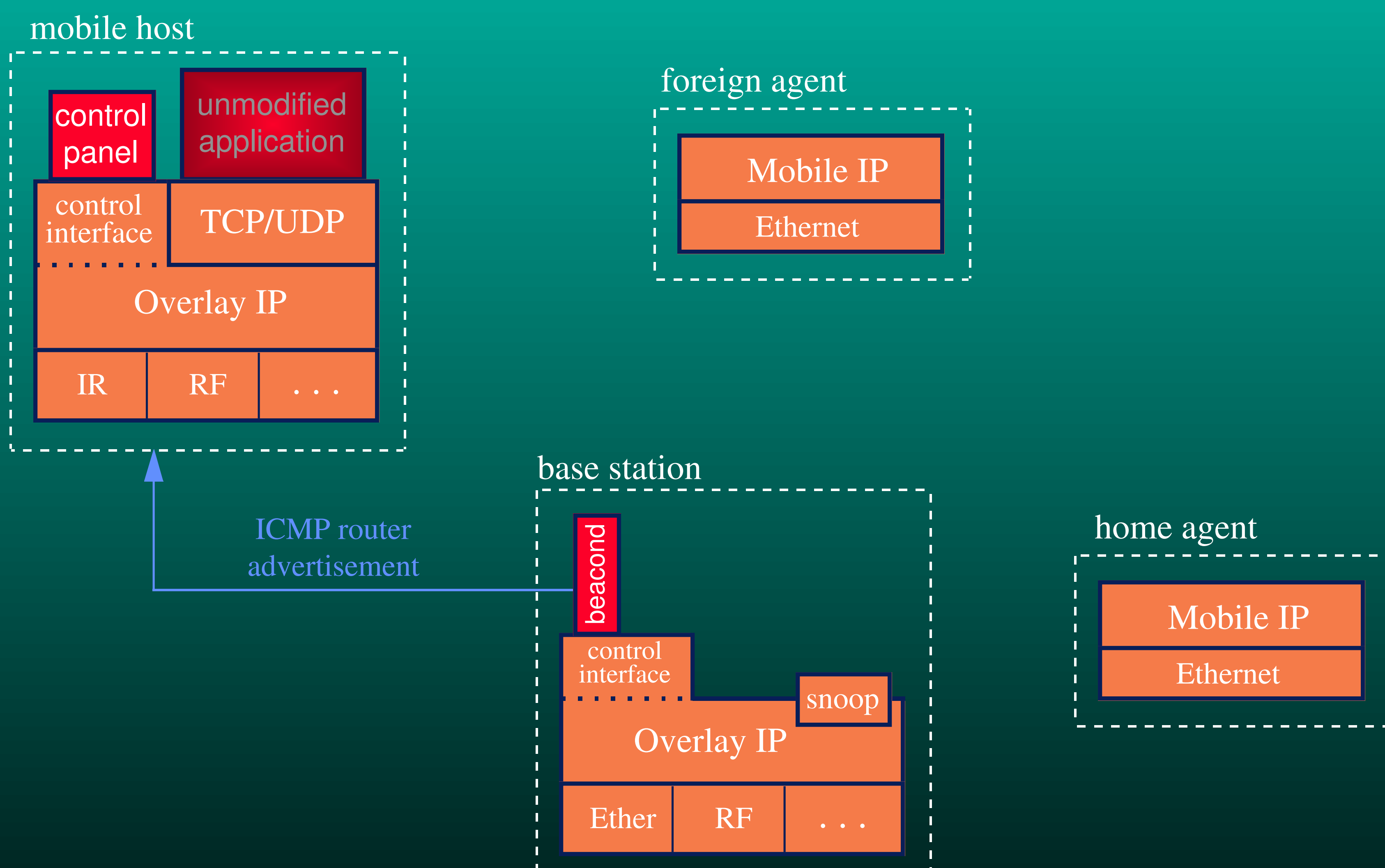
Core Proxy Architecture



Outline

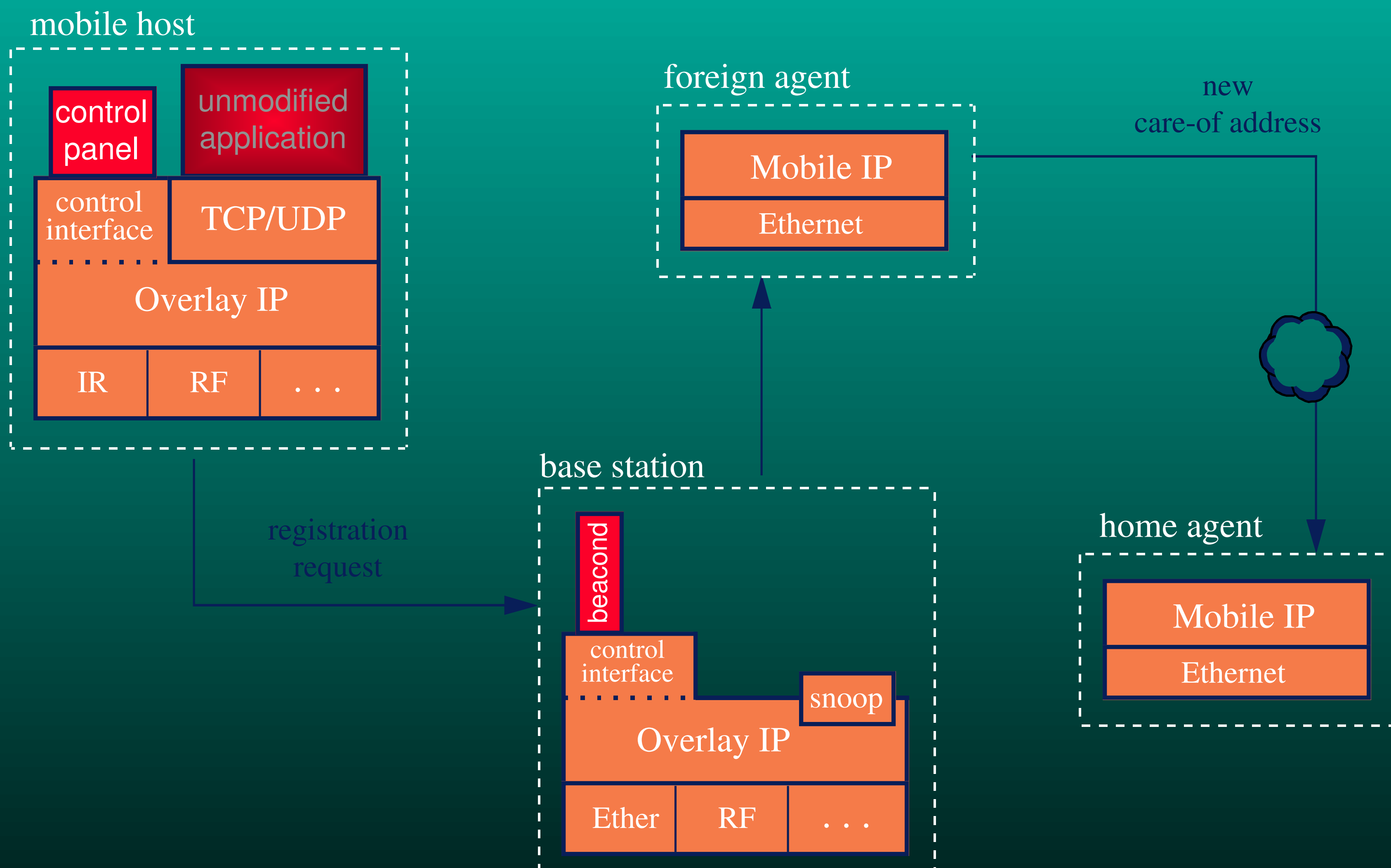
- Motivation and overview
- Core architecture components
- Core architecture walkthrough
  - connection, document retrieval, vertical handoff, document refinement
- Extended architecture components
- Extended architecture walkthrough
  - connection and authentication, vertical handoff, horizontal handoff, dynamic adaptation, load balancing, domain and network services
- Summary, Open Issues

Connecting to the Network
(core architecture - Mobile IP)



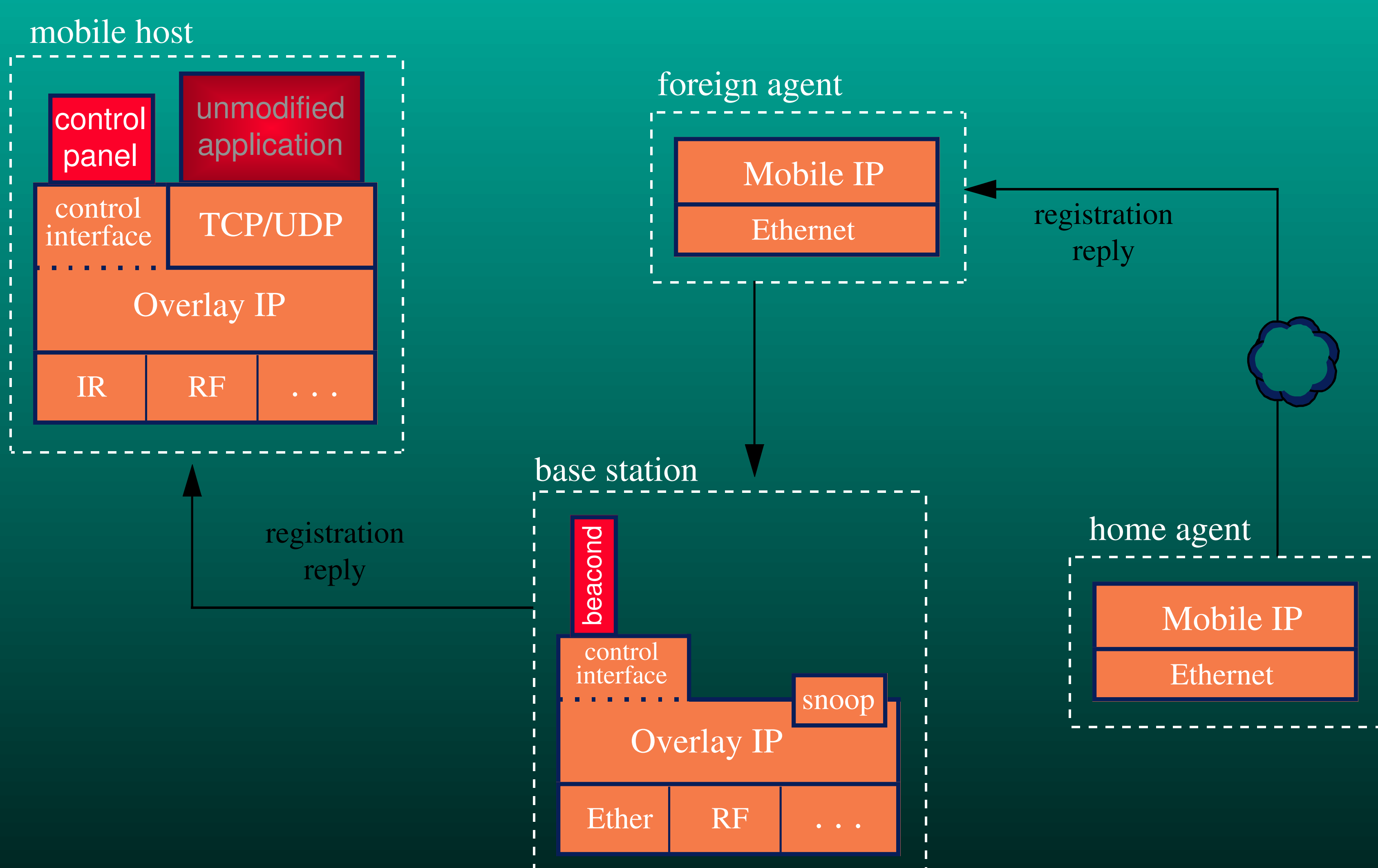
Connecting to the Network

(core architecture - Mobile IP)

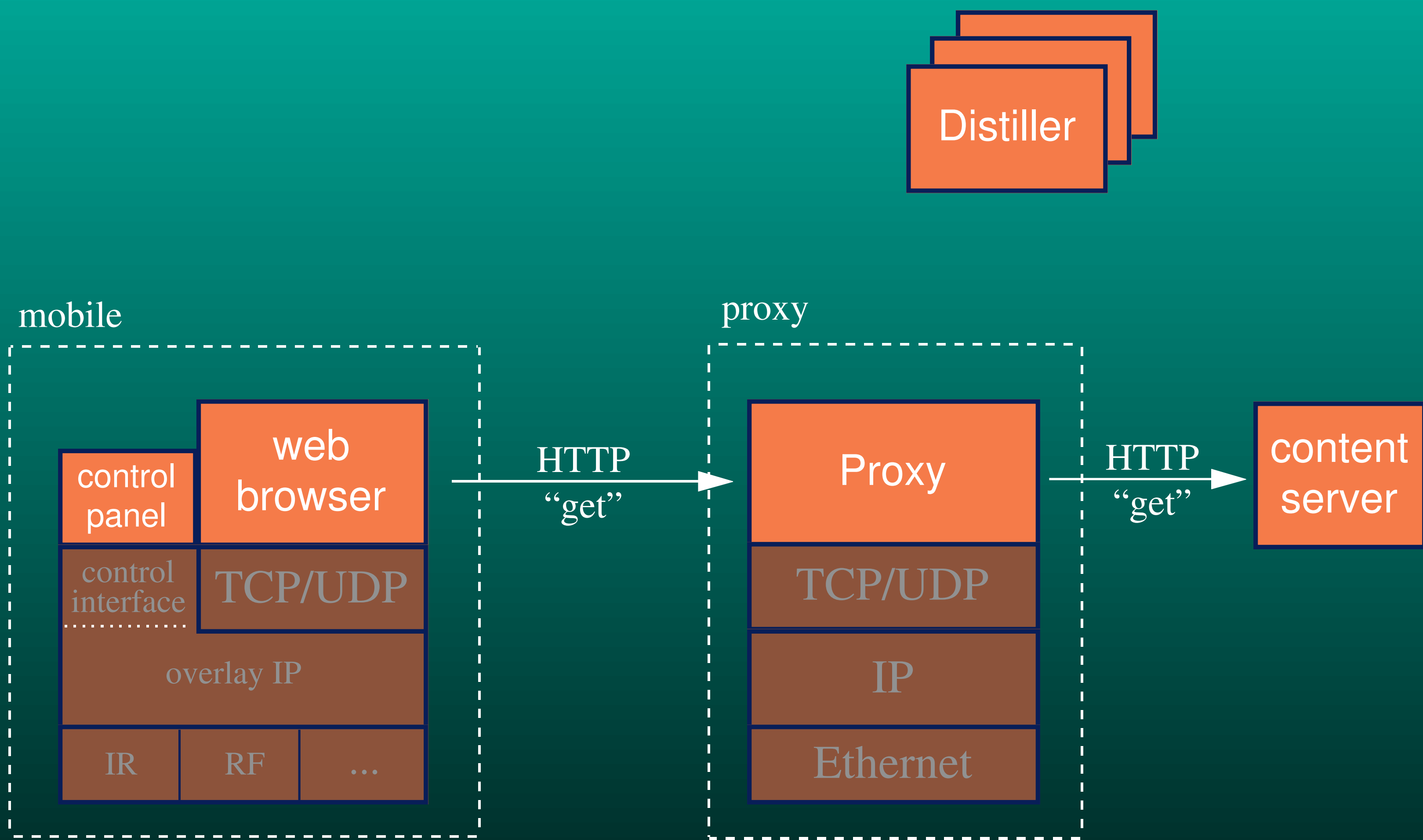


Connecting to the Network

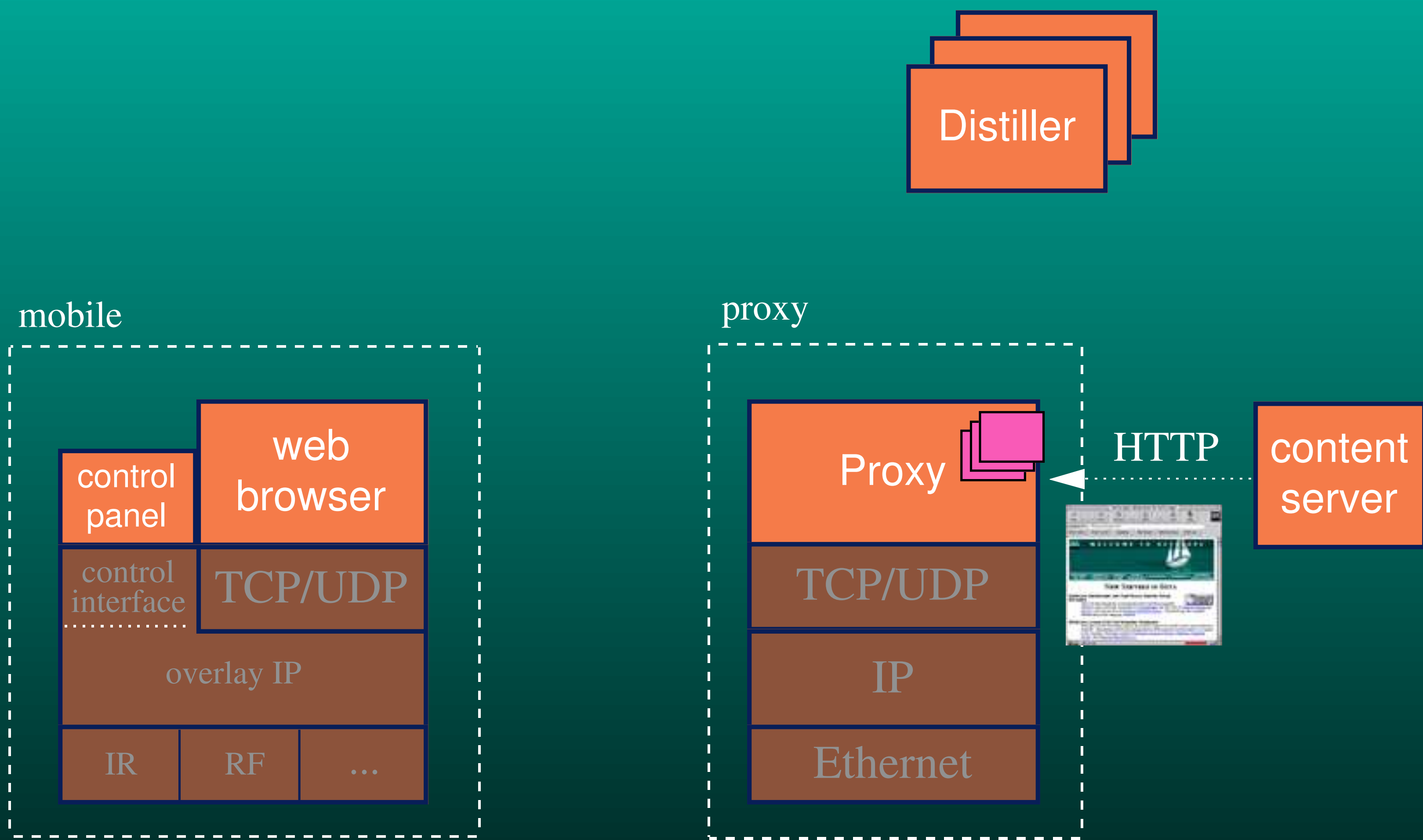
(core architecture - Mobile IP)



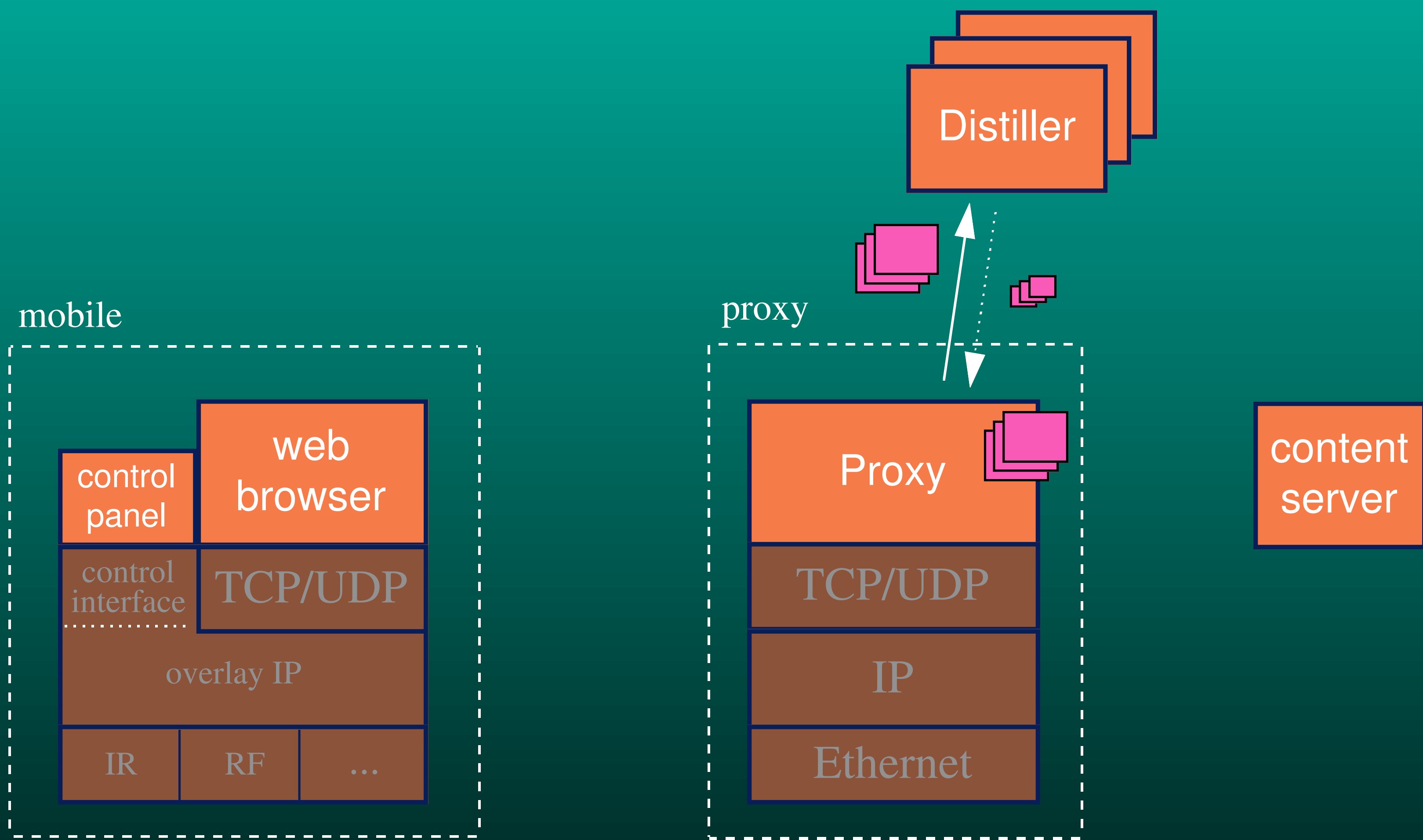
Document Request



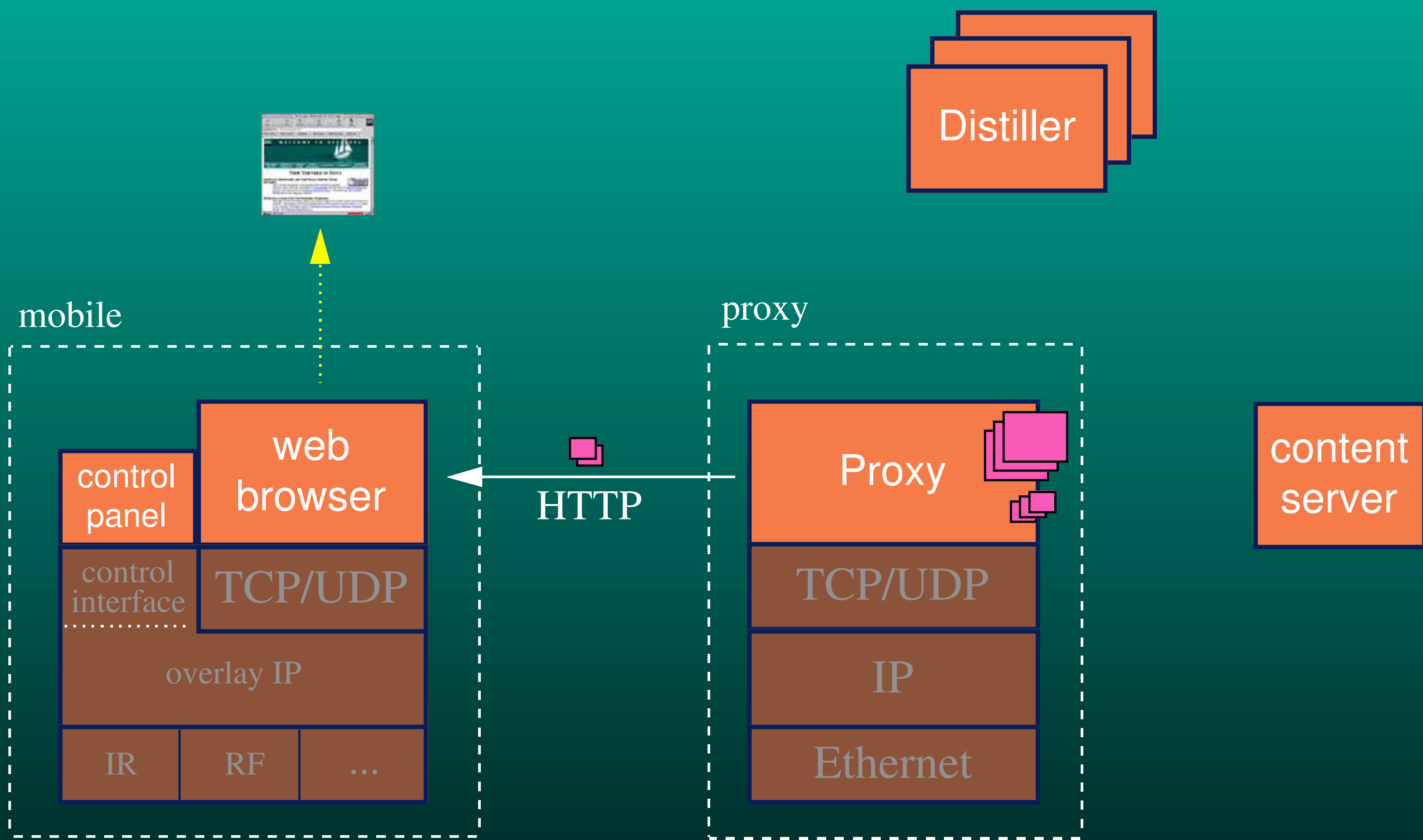
Document Download and 'Chunking'



Distillation

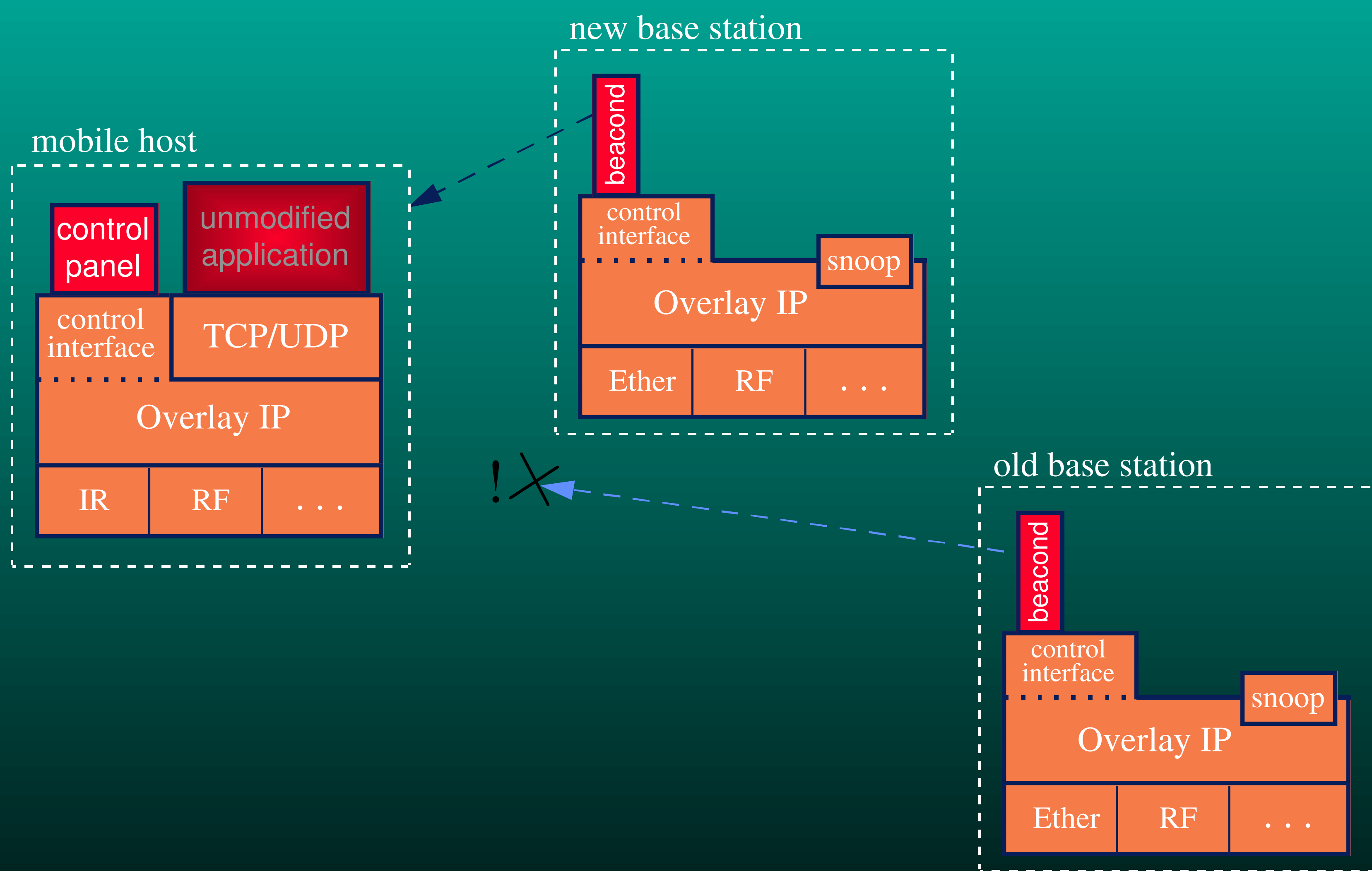


Document Upload

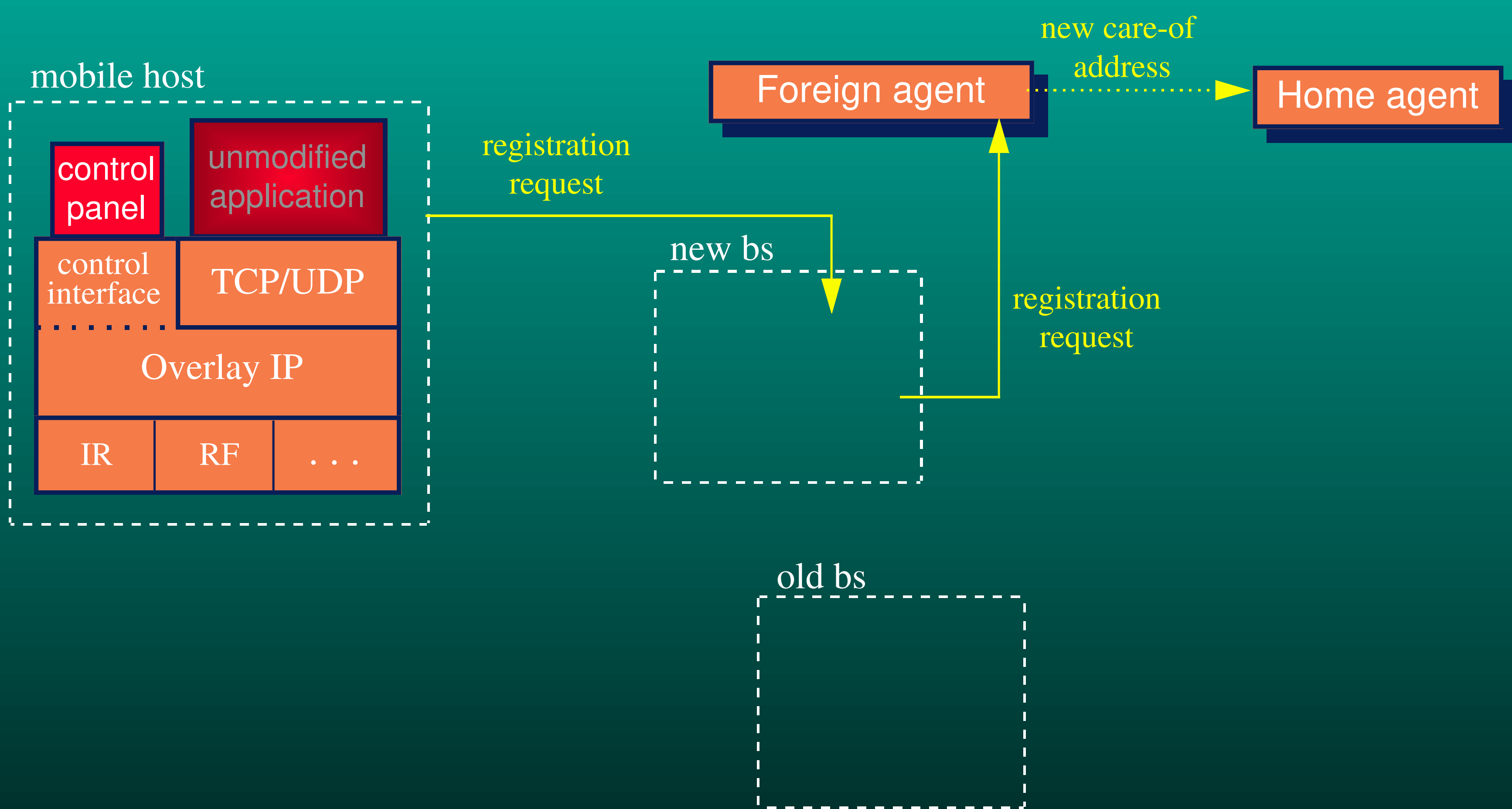


Vertical Handoff from IR to WaveLAN

(core architecture)

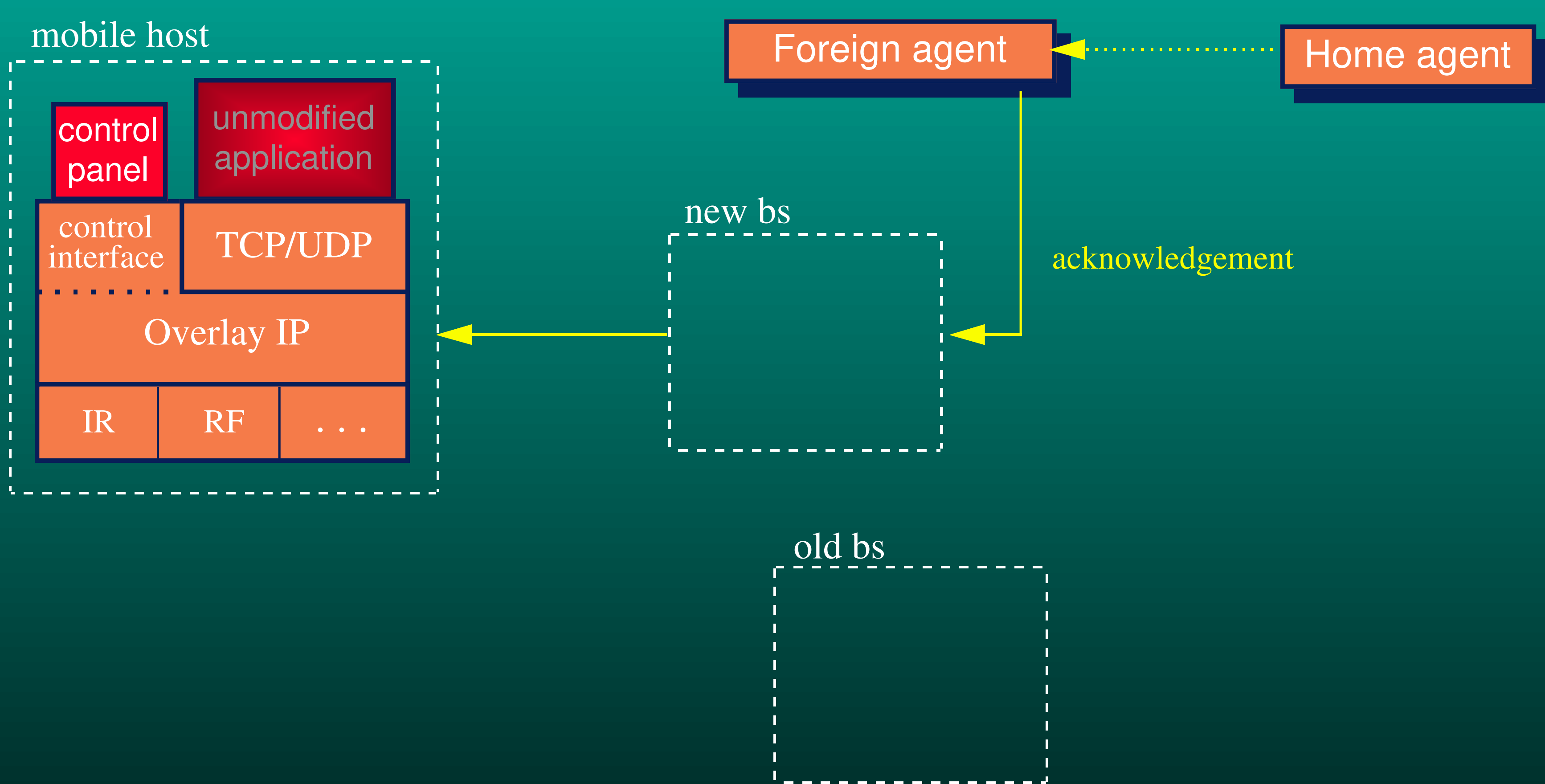


Handoff Subroutine
(core architecture)

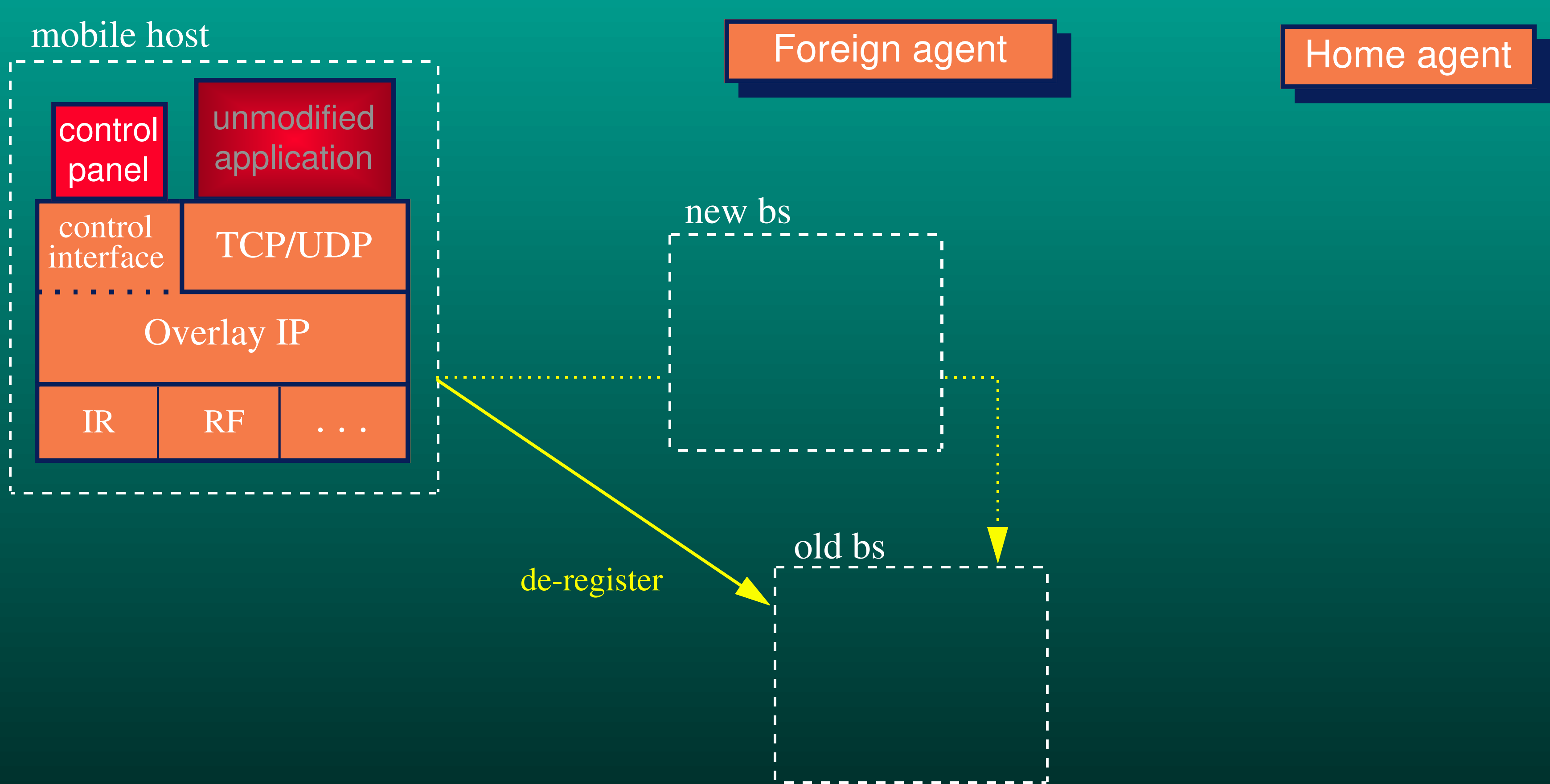


Handoff Subroutine

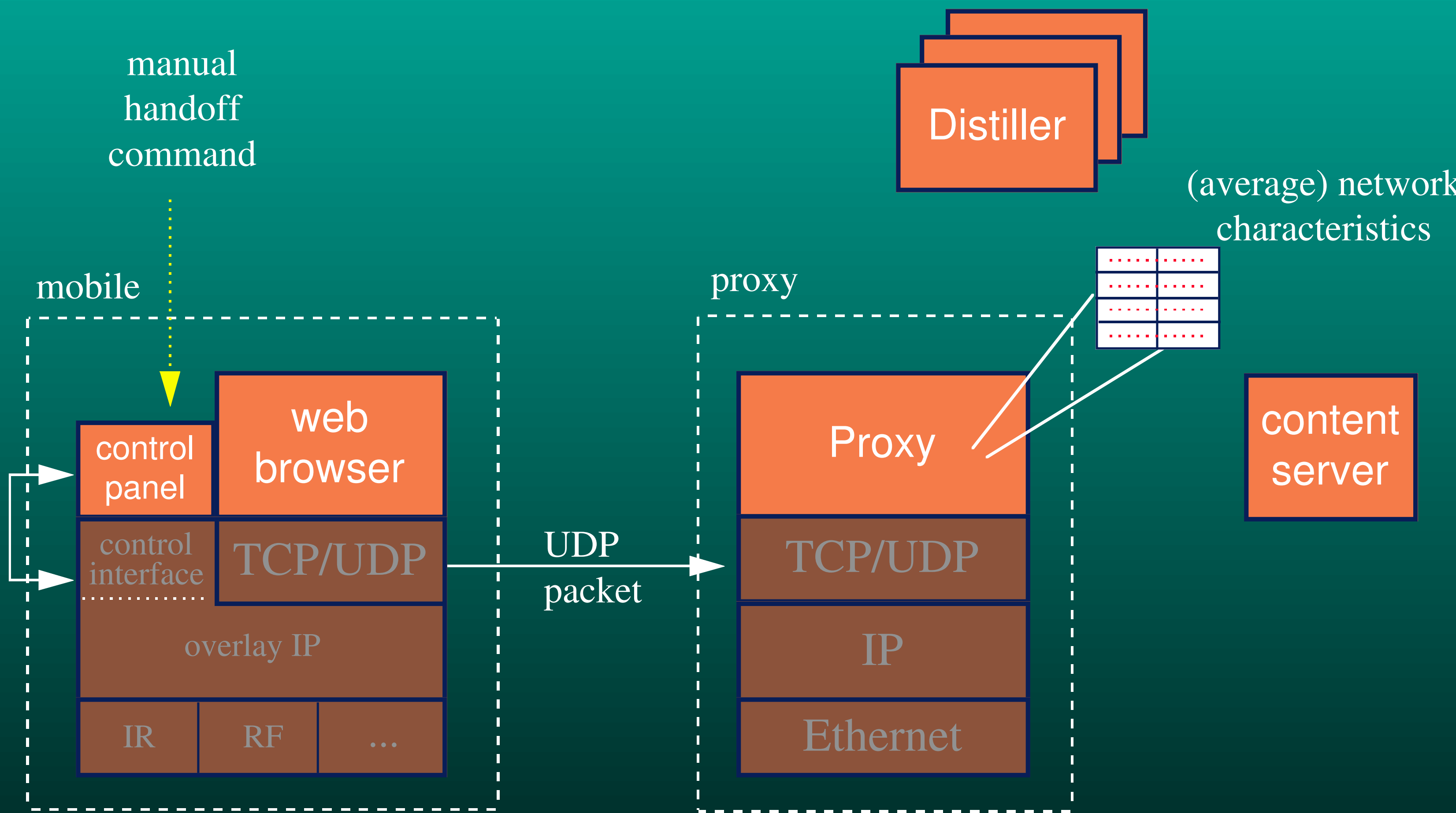
(core architecture)



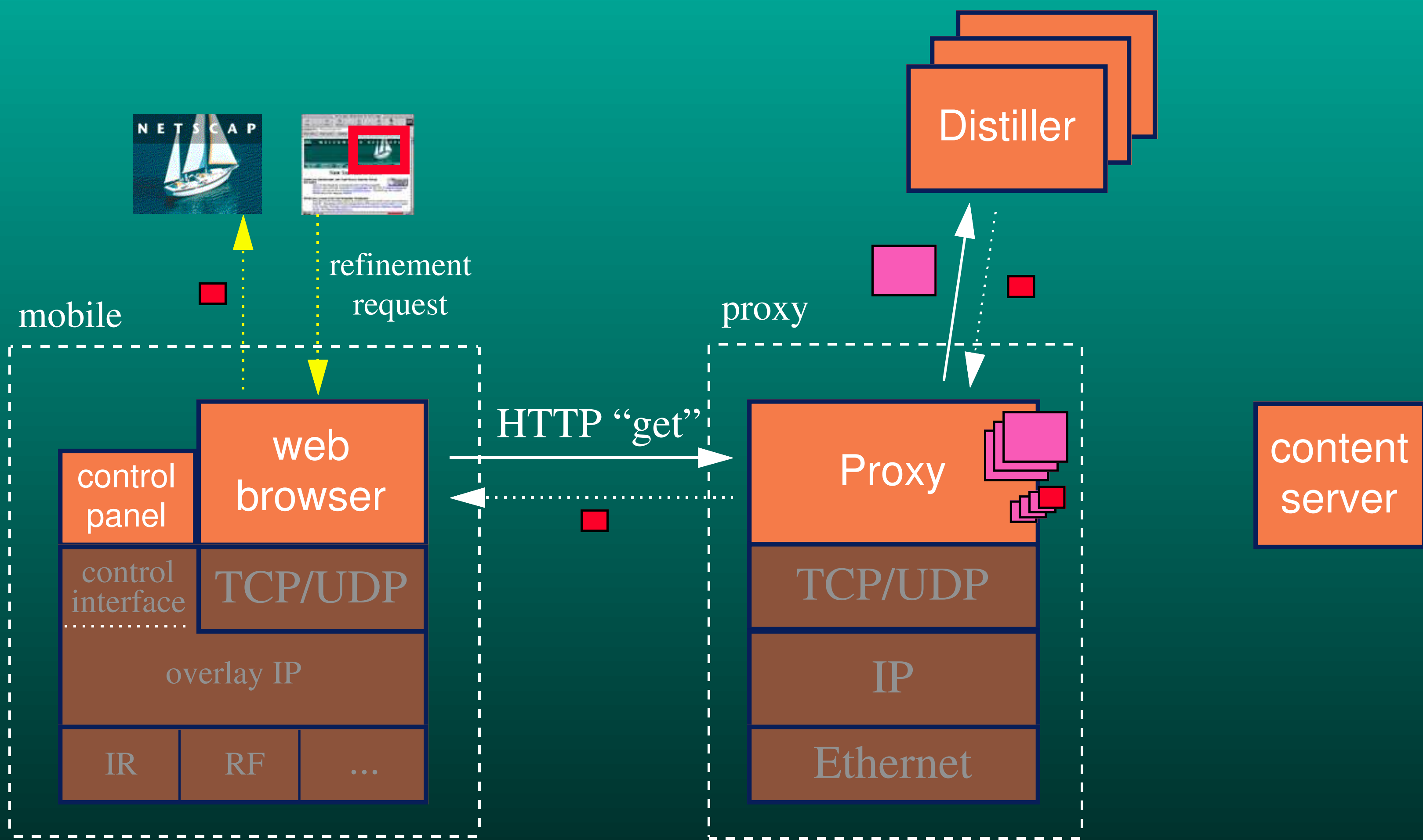
Handoff Subroutine
(core architecture)



Handoff Notification and Dynamic Adaptation



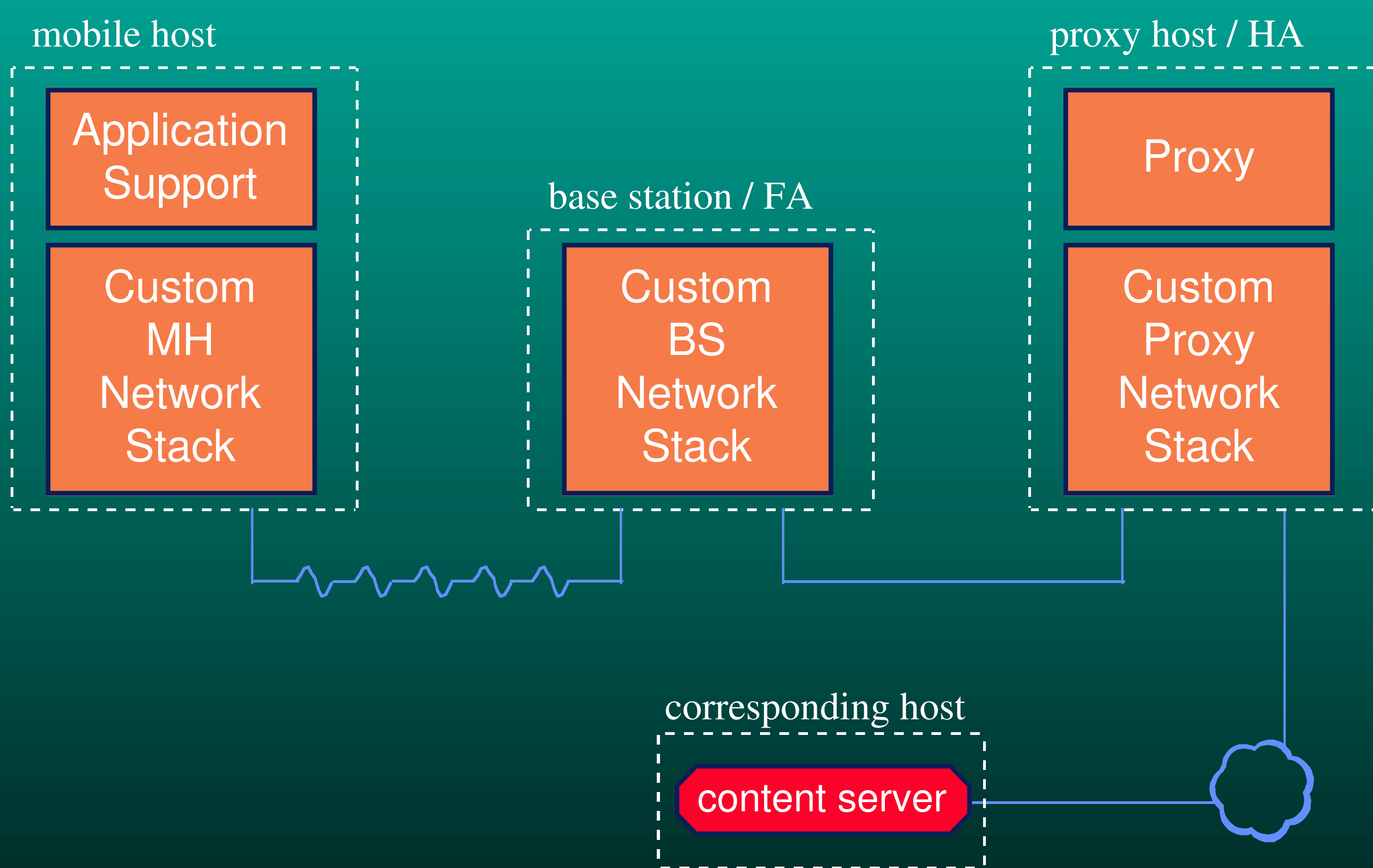
‘Chunk’ Refinement



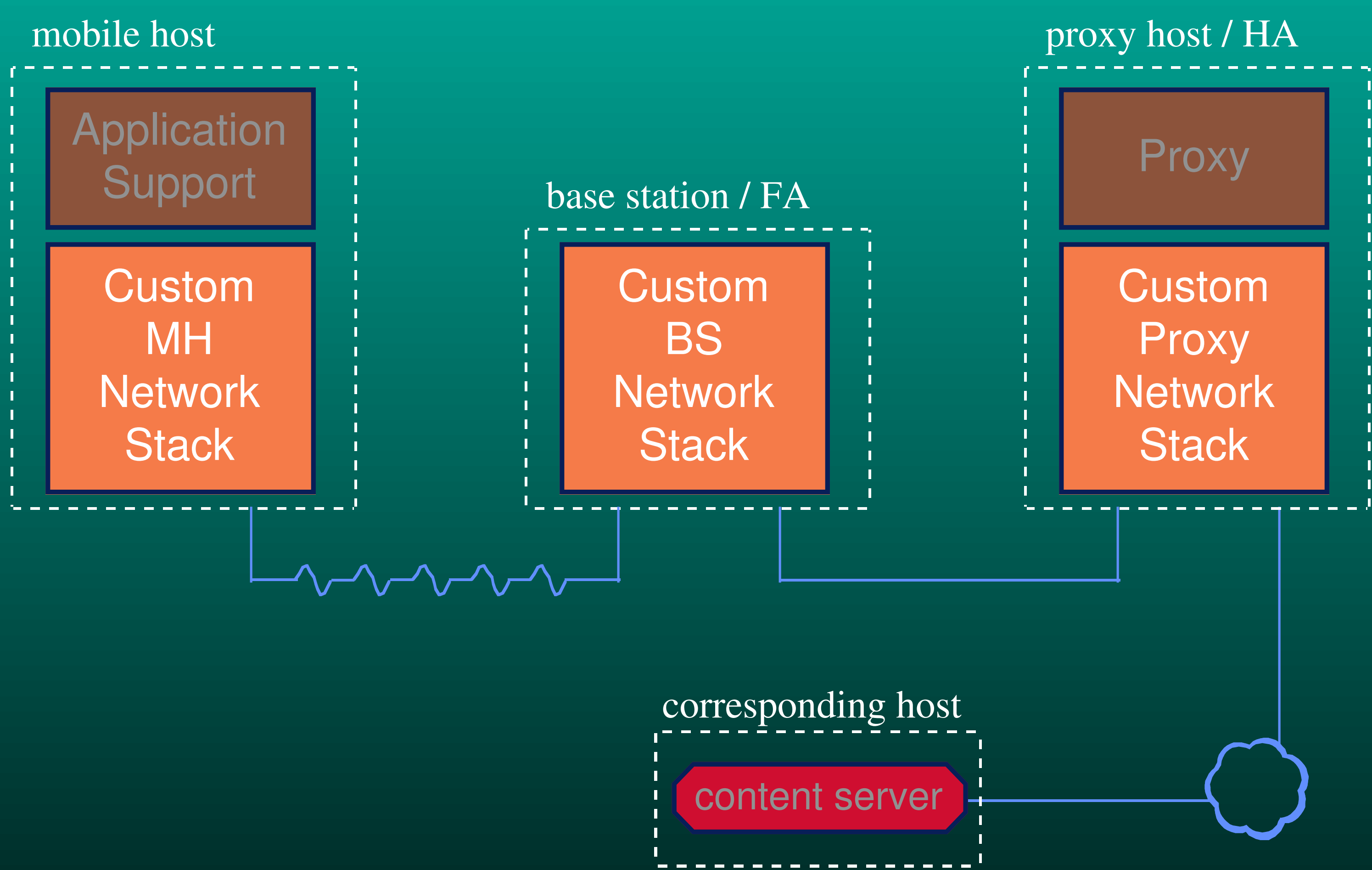
Outline

- Motivation and overview
- Core architecture components
- Core architecture walkthrough
  - connection, document retrieval, vertical handoff, document refinement
- Extended architecture components
- Extended architecture walkthrough
  - connection and authentication, vertical handoff, horizontal handoff, dynamic adaptation, load balancing, domain and network services
- Summary, Open Issues

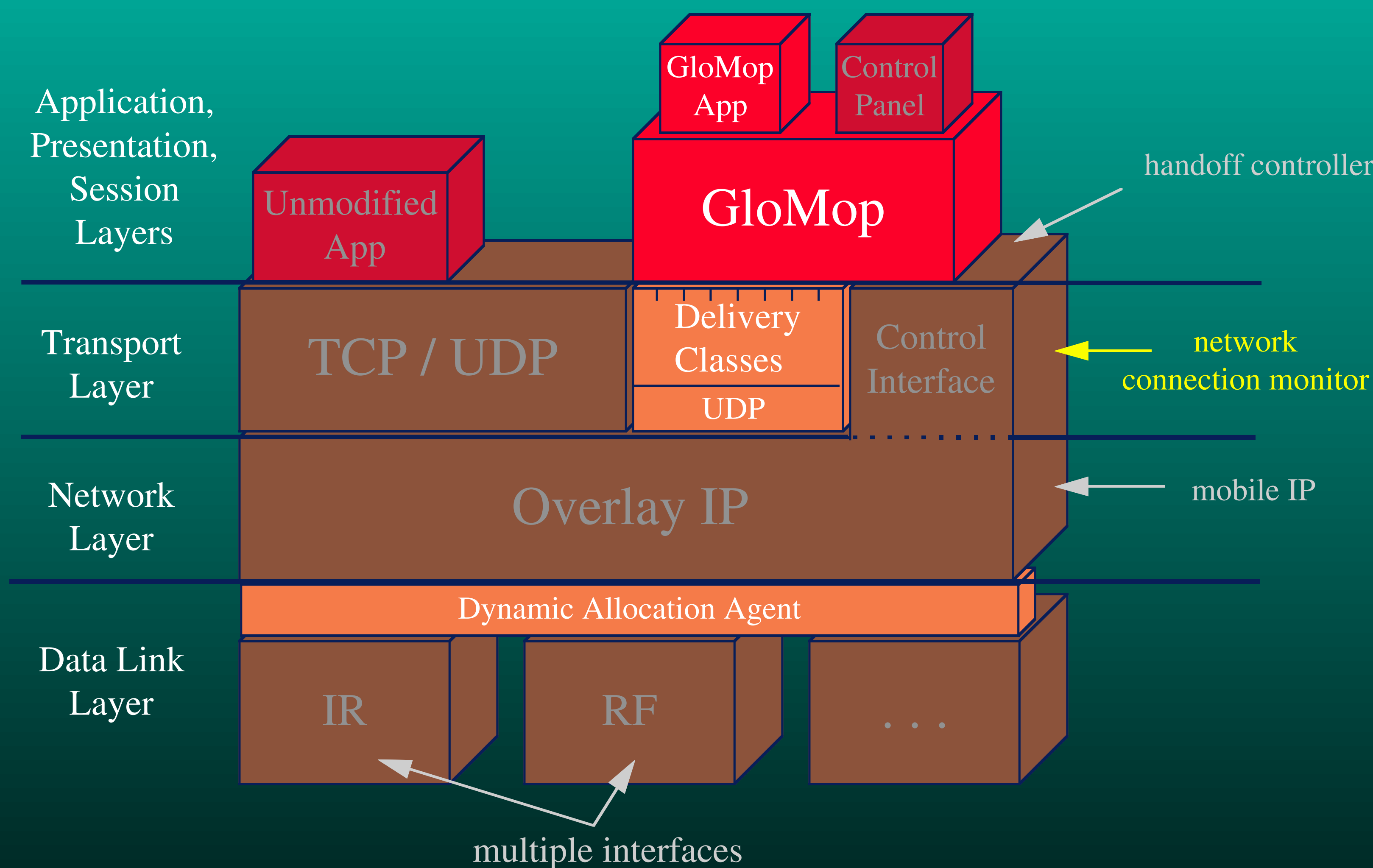
Extended Architecture Components



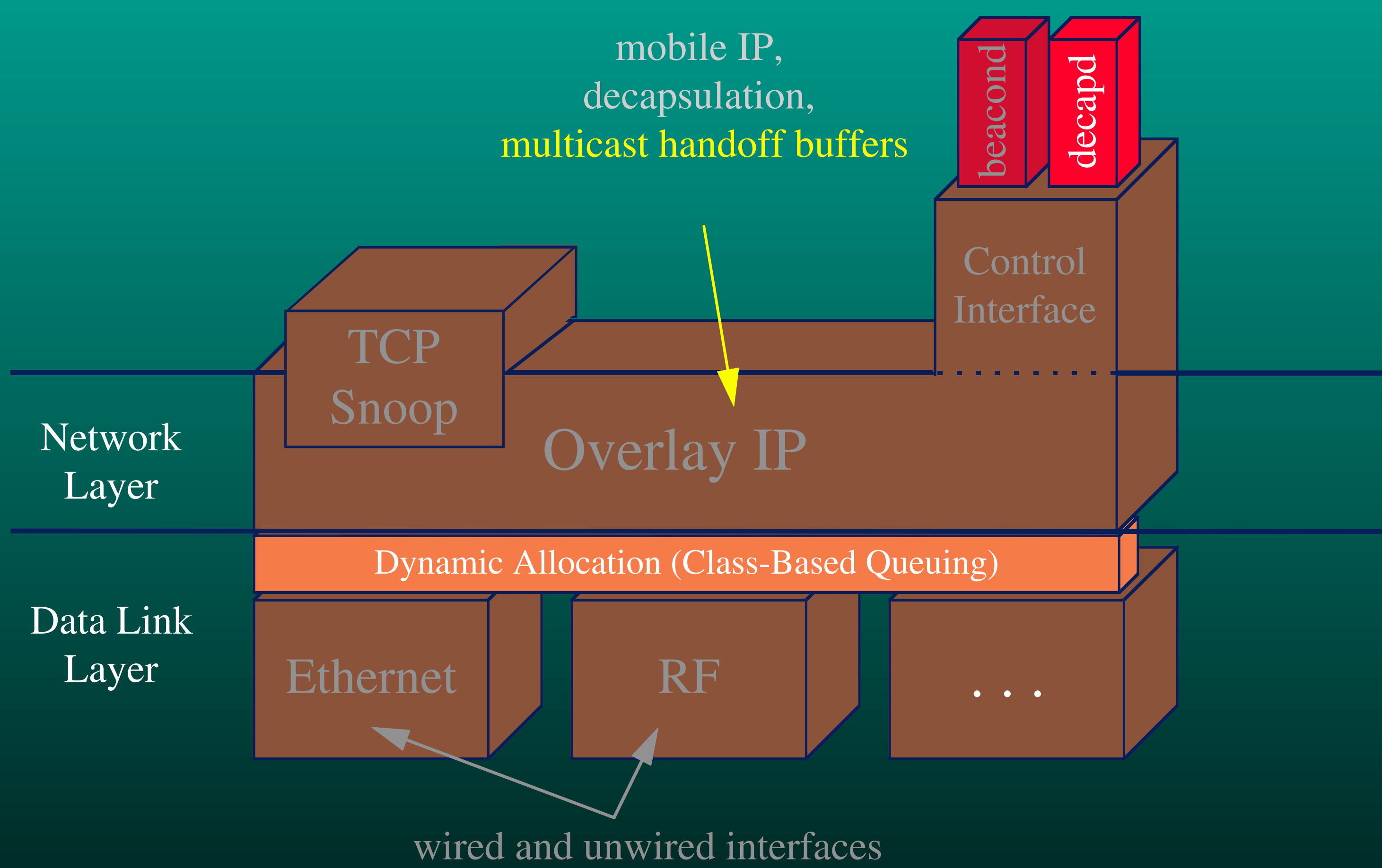
Extended Architecture Components



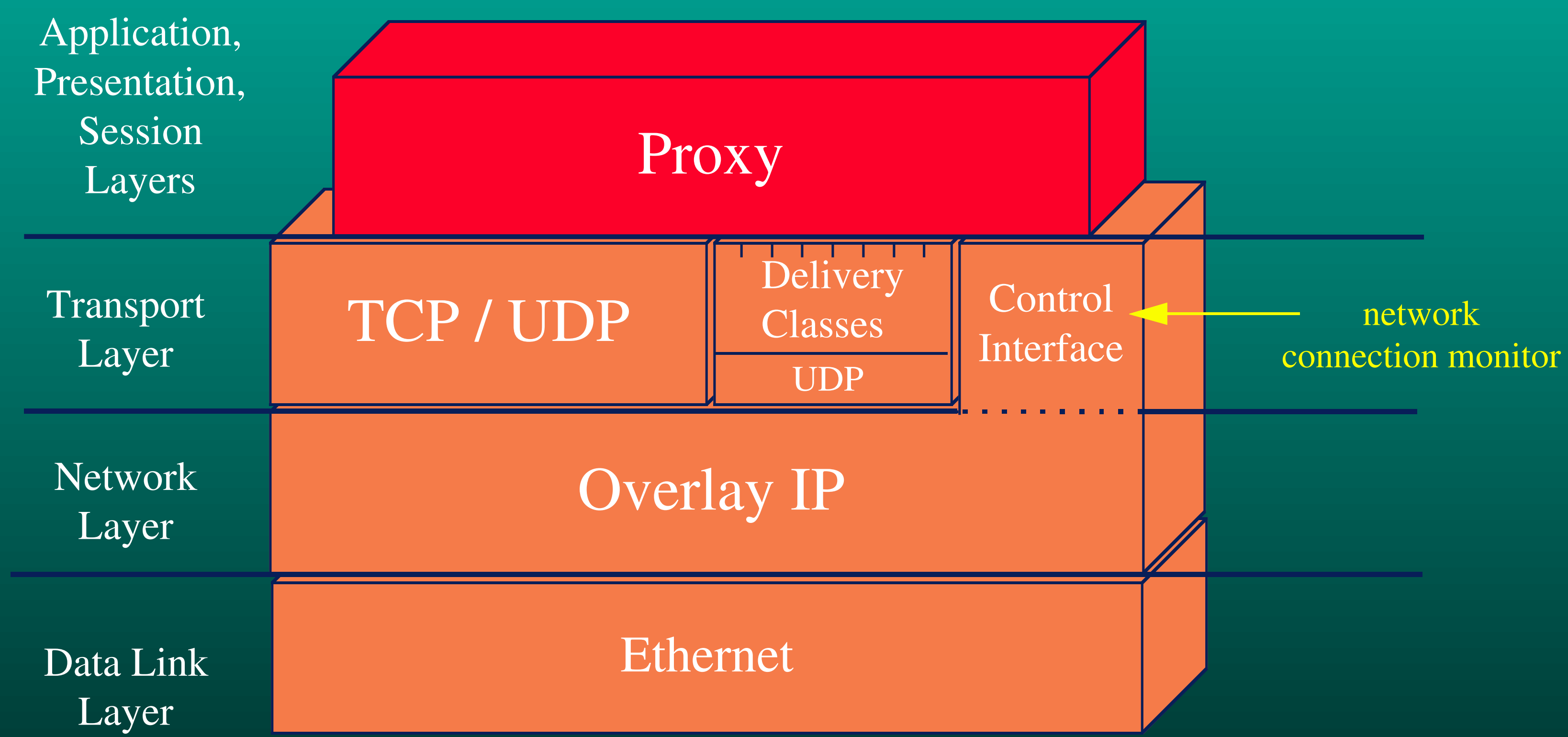
Extended Daedalus Mobile Host



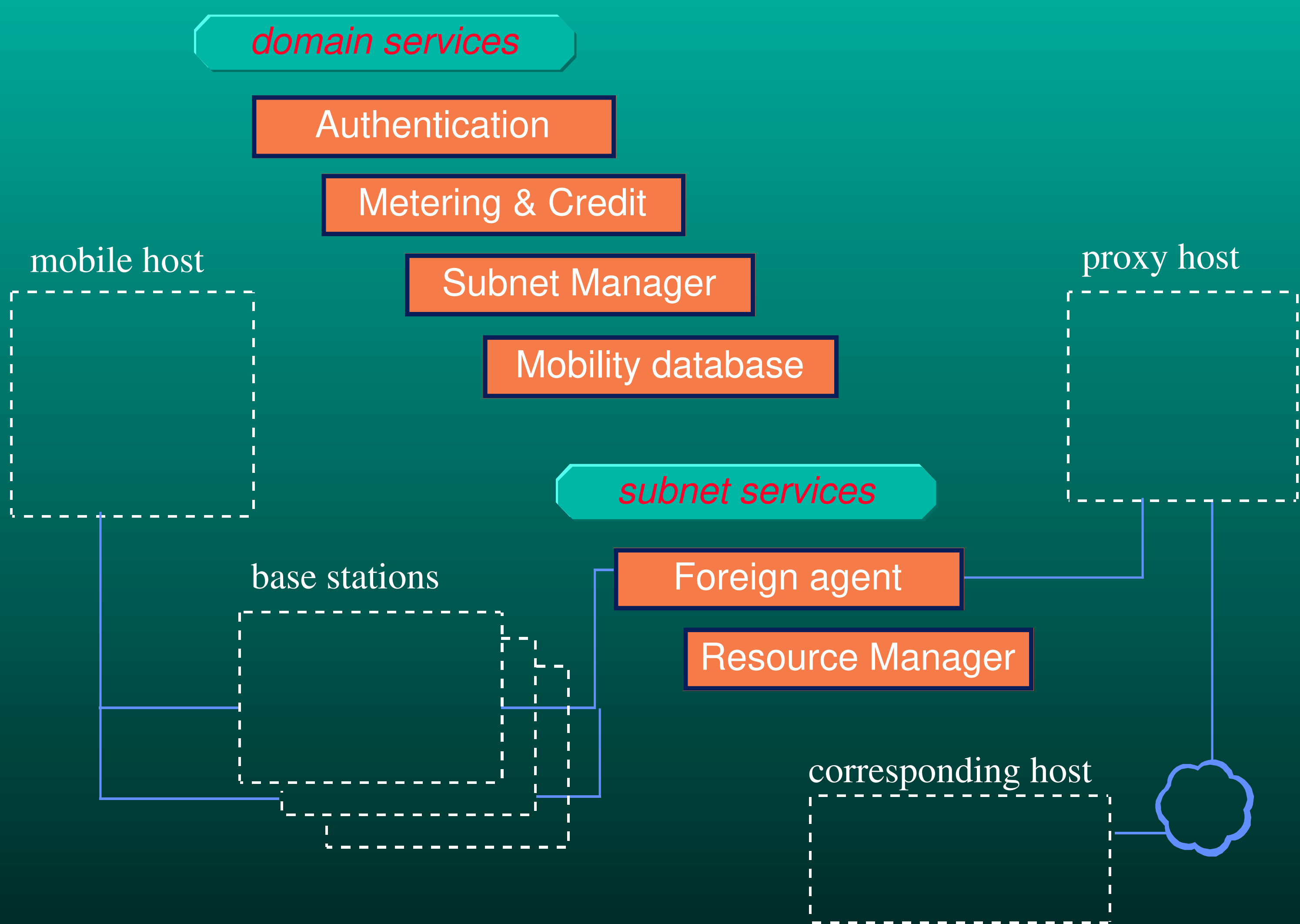
Extended Daedalus Base Station



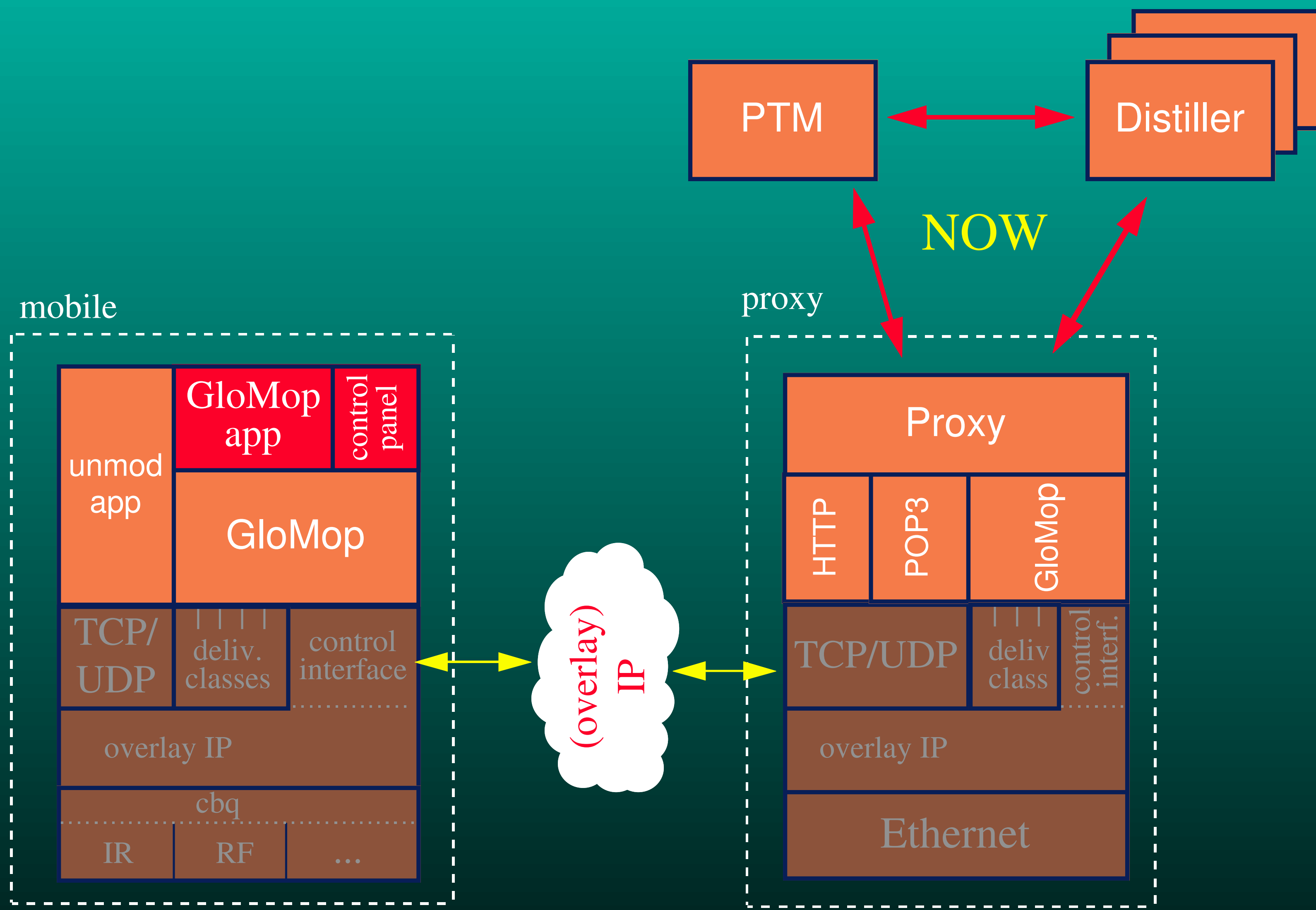
Extended Daedalus Proxy



Extended Service Architecture



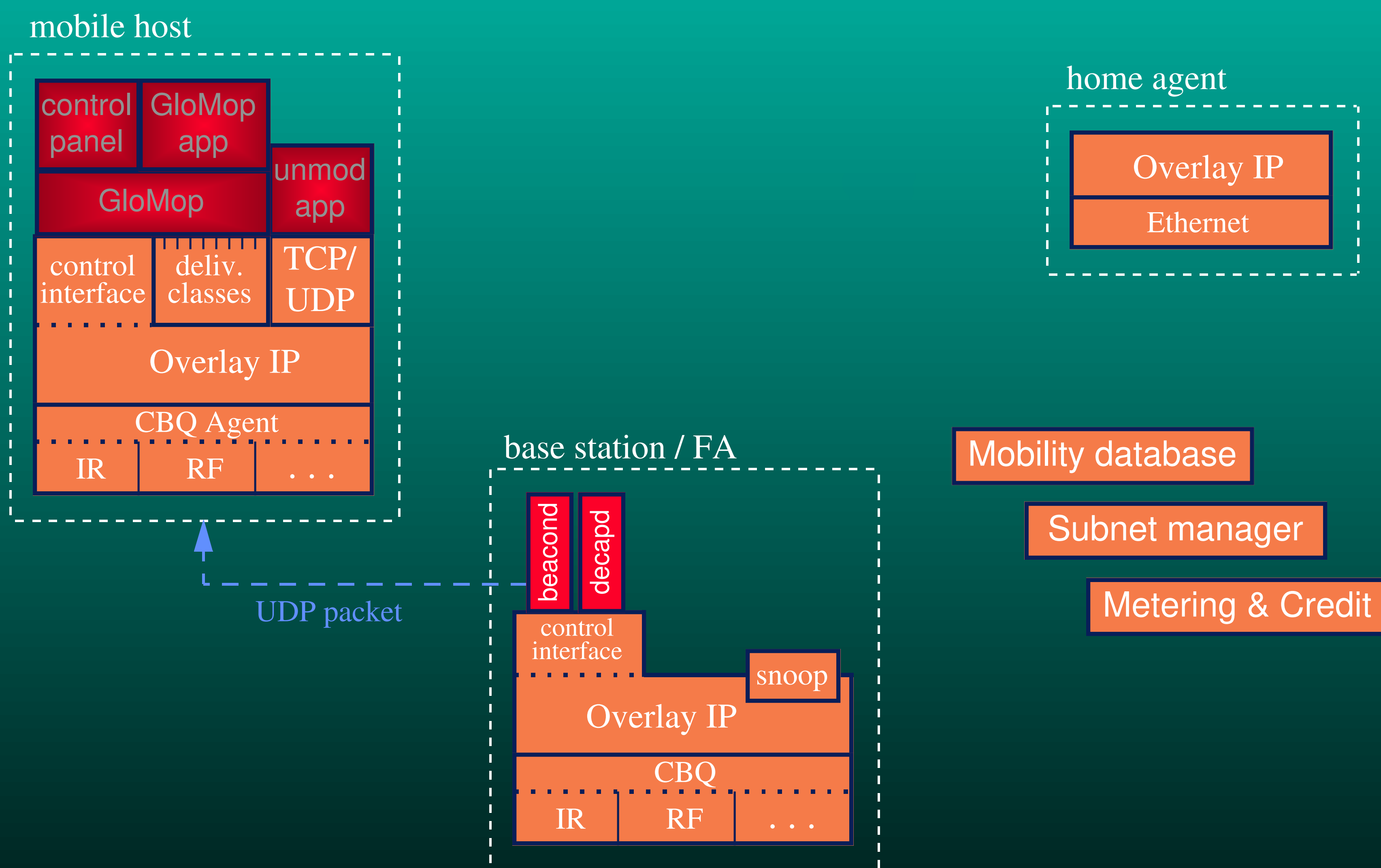
Extended proxy architecture



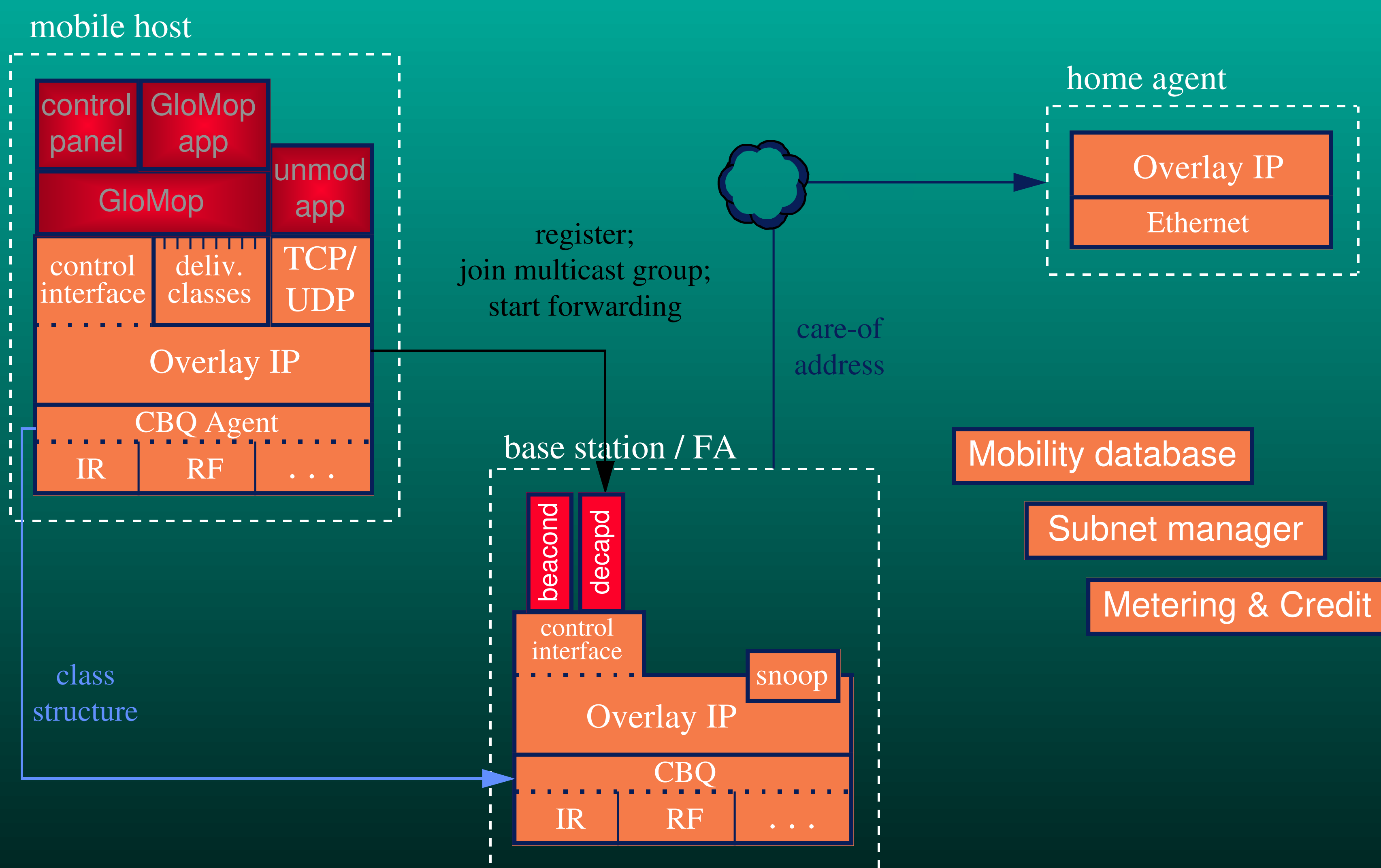
Outline

- Motivation and overview
- Core architecture components
- Core architecture walkthrough
  - connection, document retrieval, vertical handoff, document refinement
- Extended architecture components
- Extended architecture walkthrough
  - connection and authentication, vertical handoff, horizontal handoff, dynamic adaptation, load balancing, domain and network services
- Summary, Open Issues

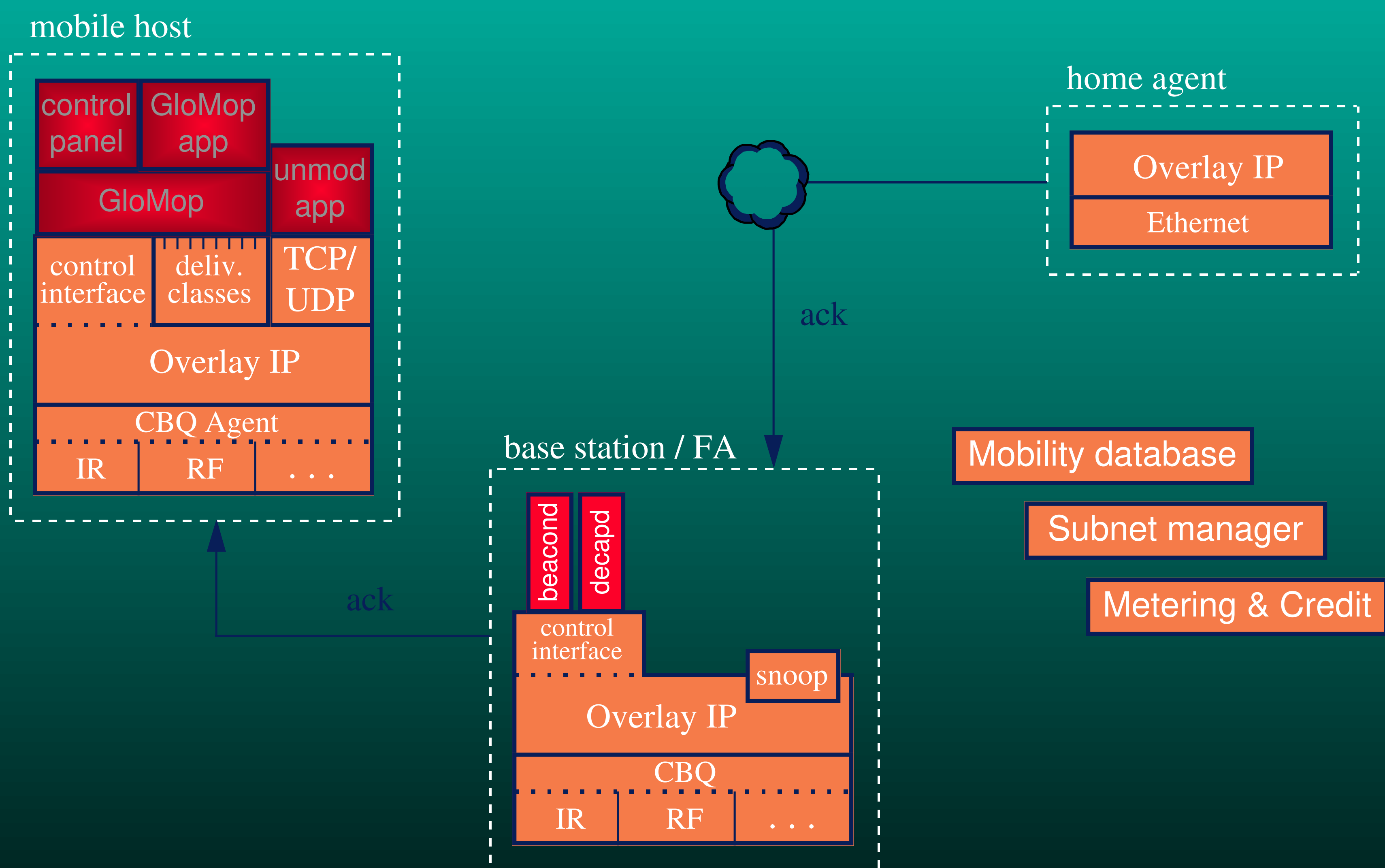
Connecting to the Network
(Multicast)



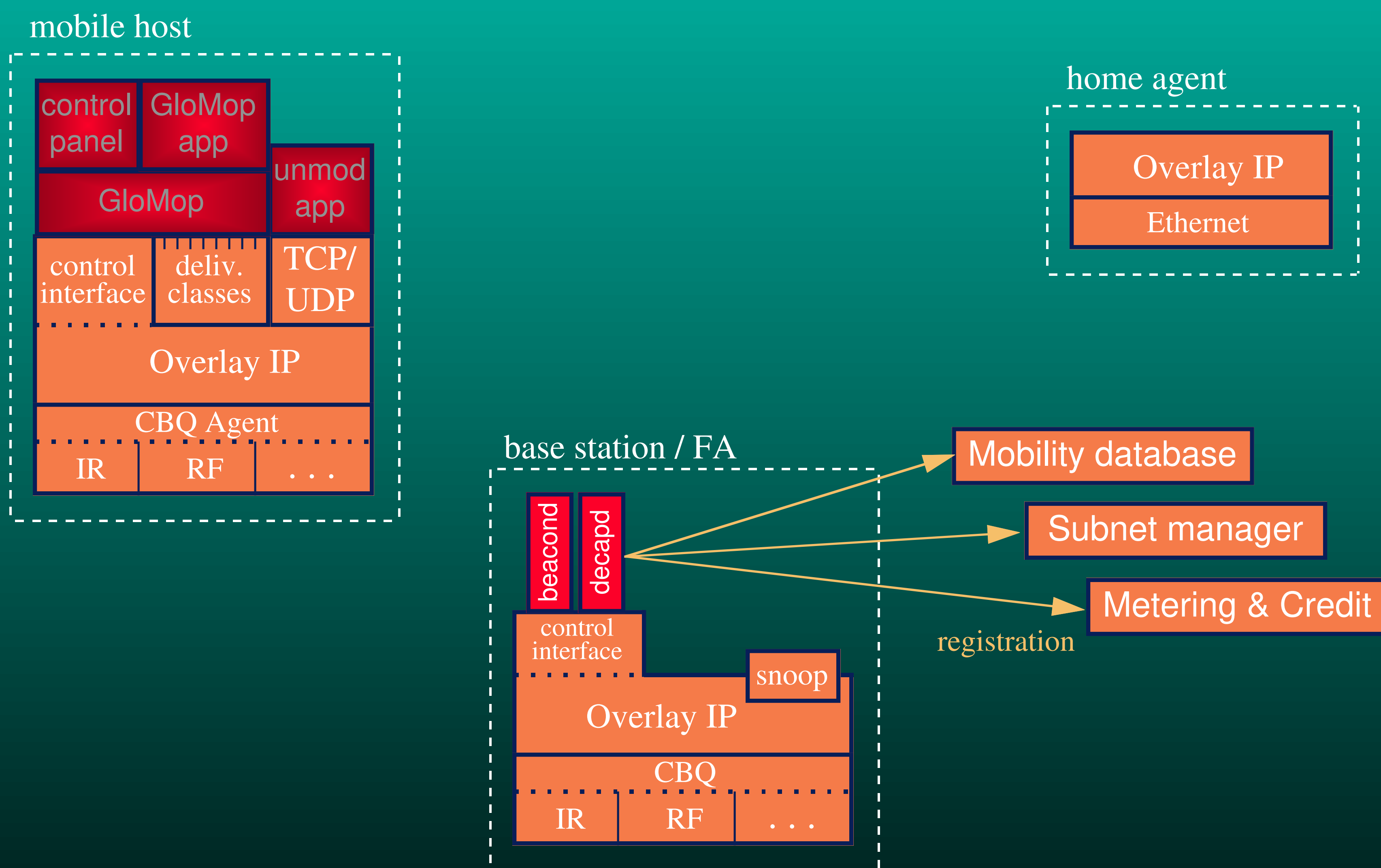
Connecting to the Network
(Multicast)



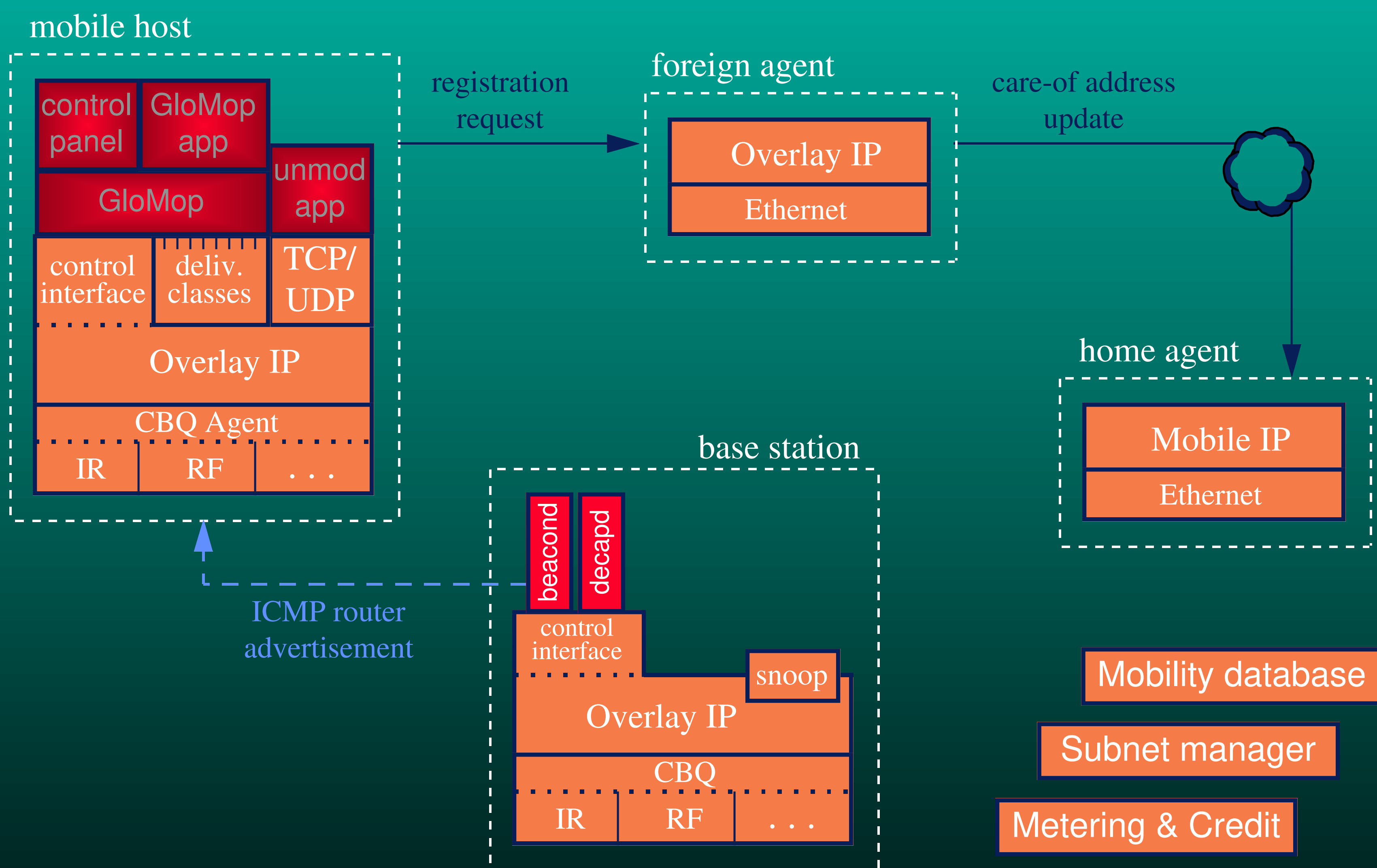
Connecting to the Network
(Multicast)



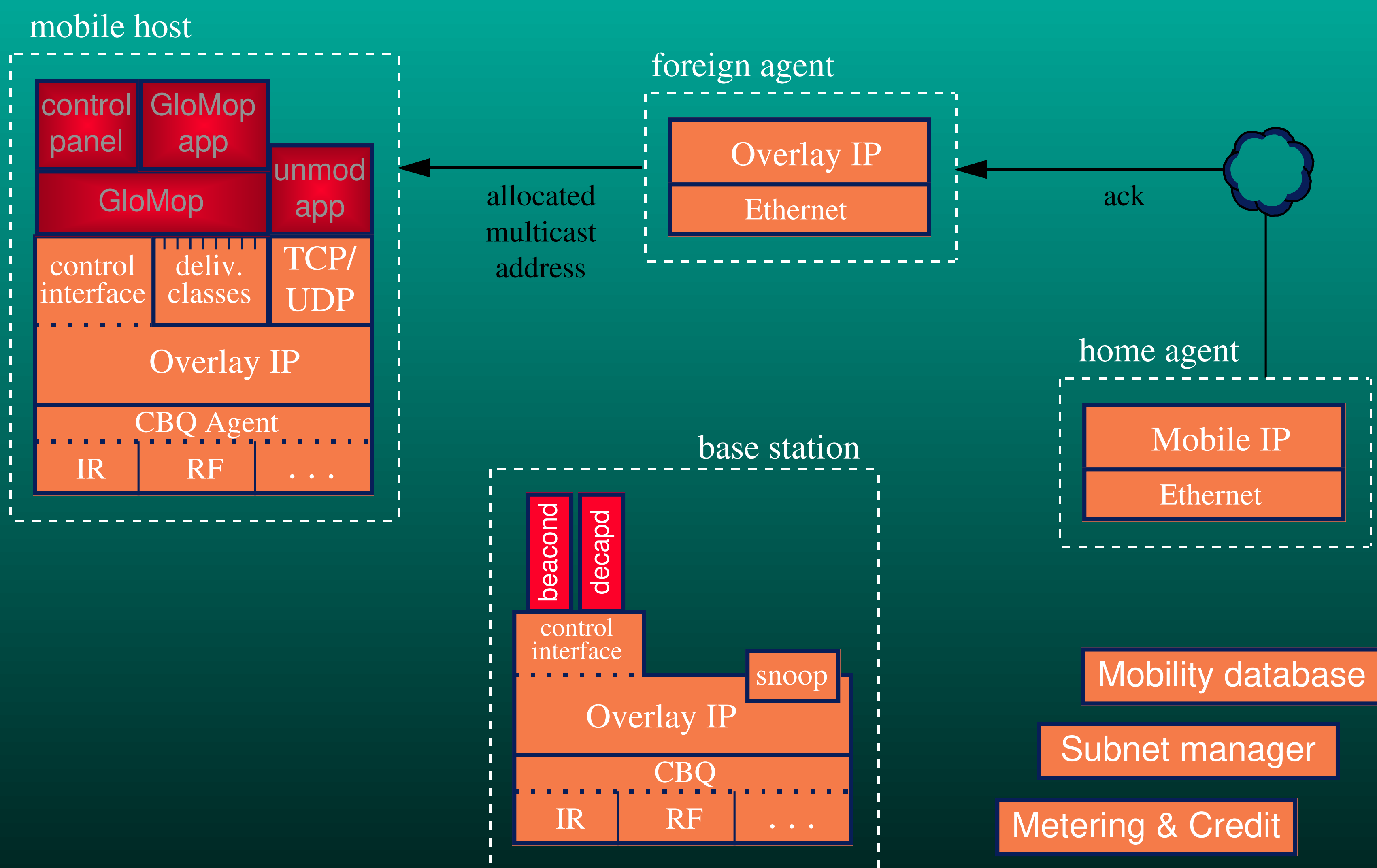
Connecting to the Network
(Multicast)



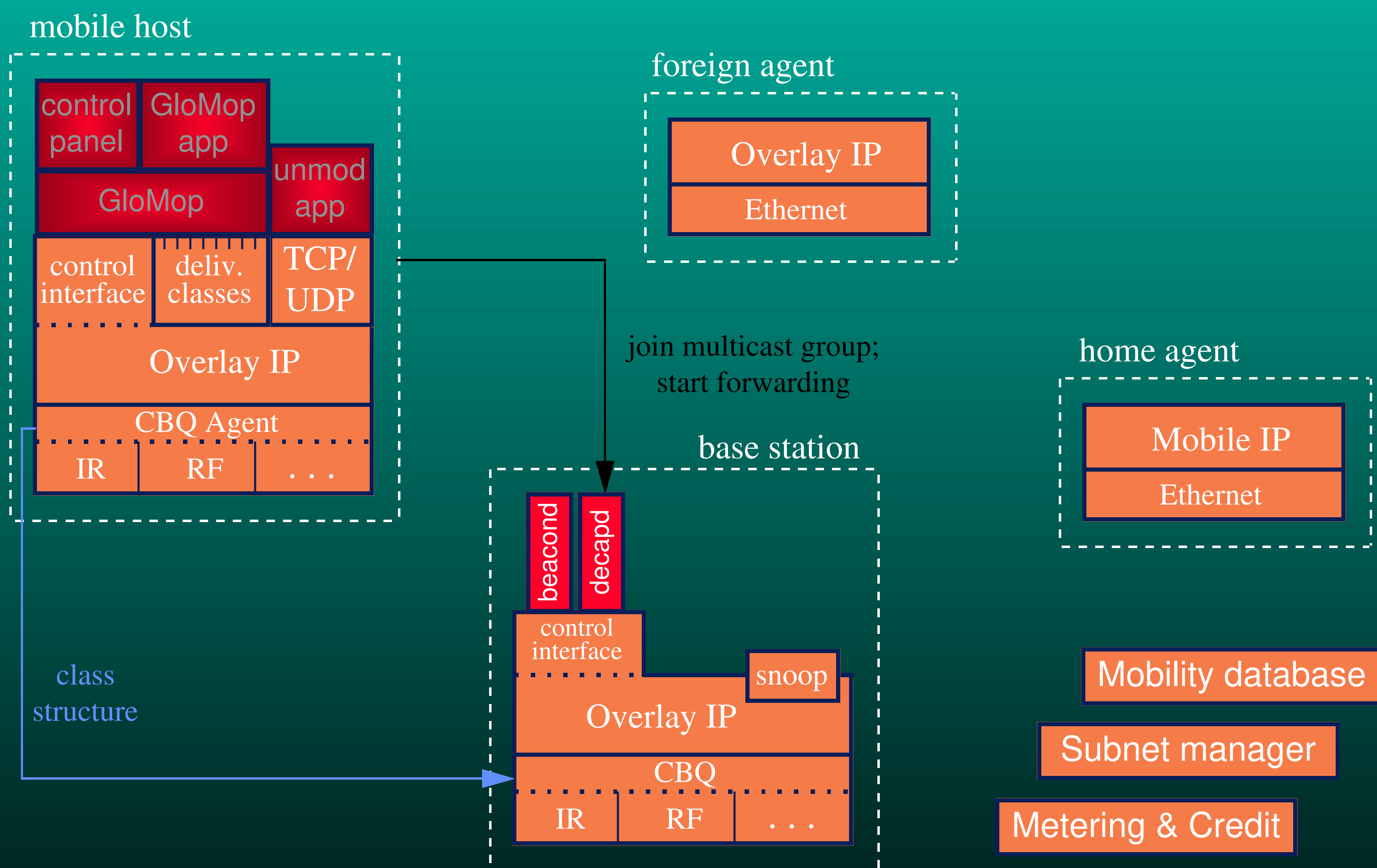
Connecting to the Network
(Hybrid)



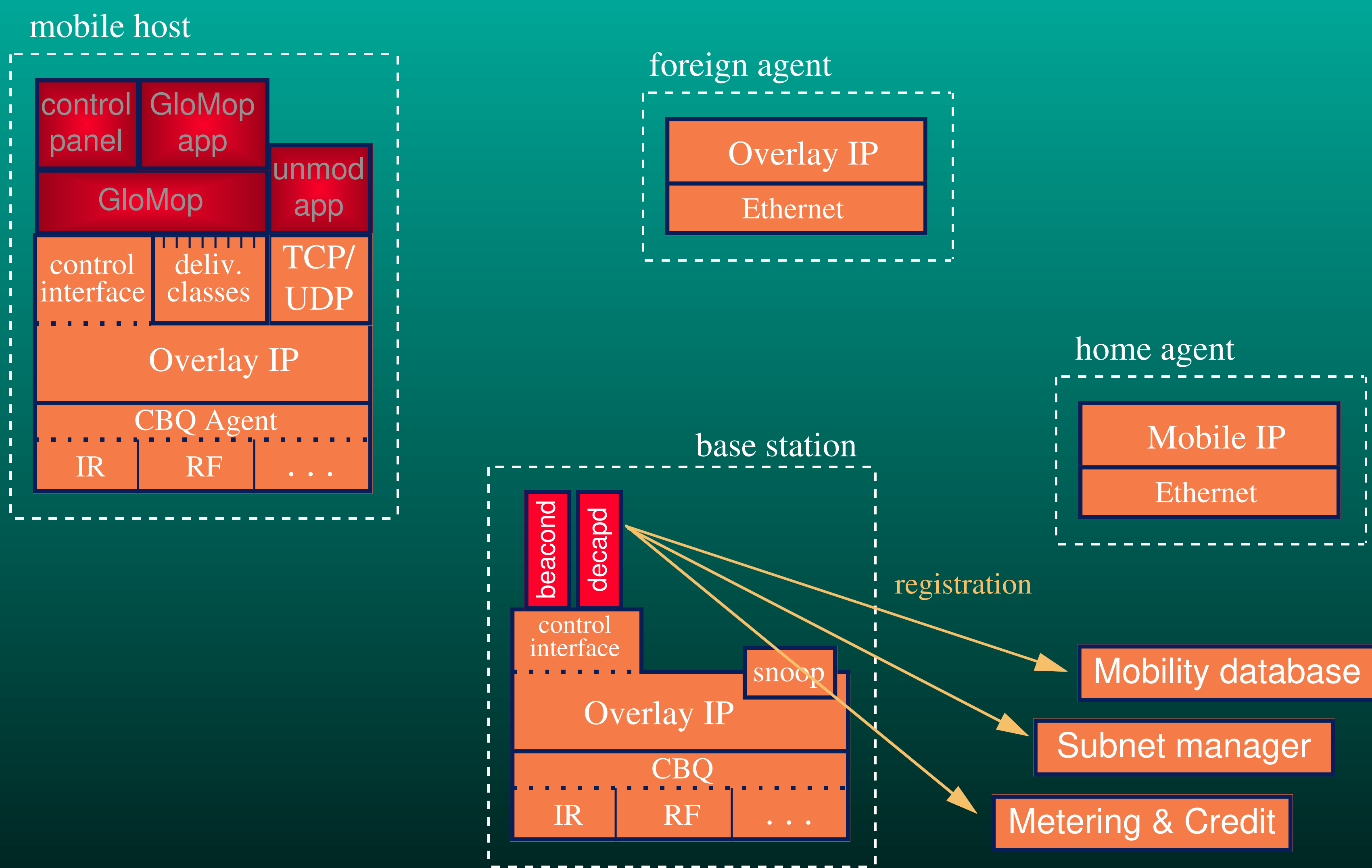
Connecting to the Network
(Hybrid)



Connecting to the Network
(Hybrid)



Connecting to the Network
(Hybrid)



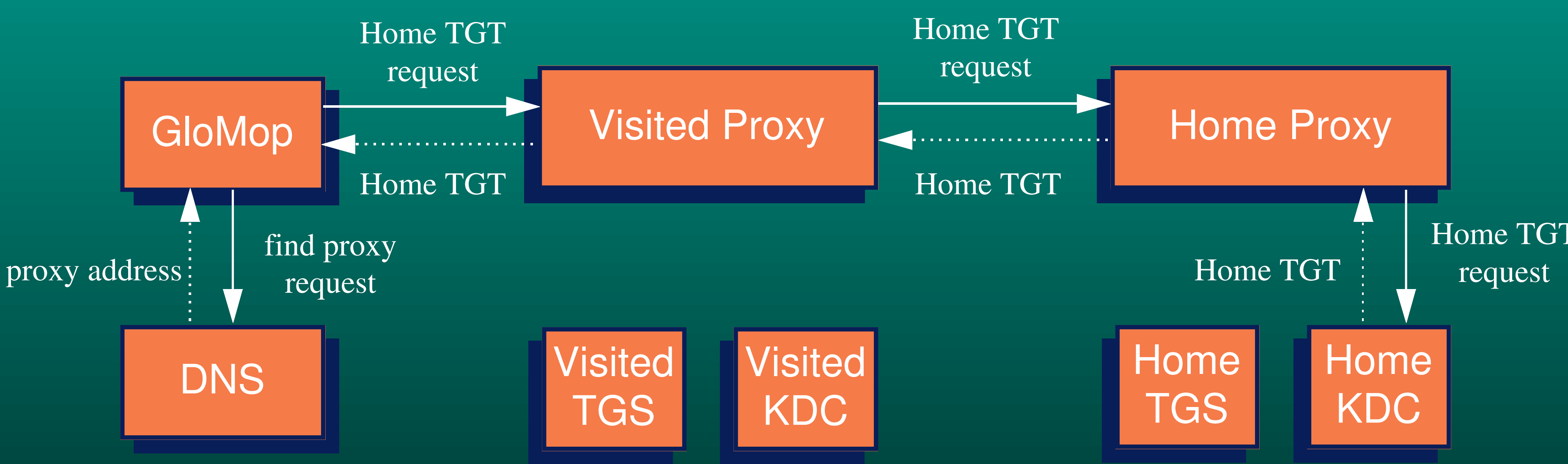
Kerberos Authentication

- centralized, secure database (KDC) holds secrets (DES key) only known by principles
- to access Kerberized services, principles are issued tickets for those services by the ticket-granting server (TGS)
- a ticket for the ticket-granting server (TGT) is issued by the KDC
- the TGT is encrypted with the issuee’s secret
- tickets contain session keys

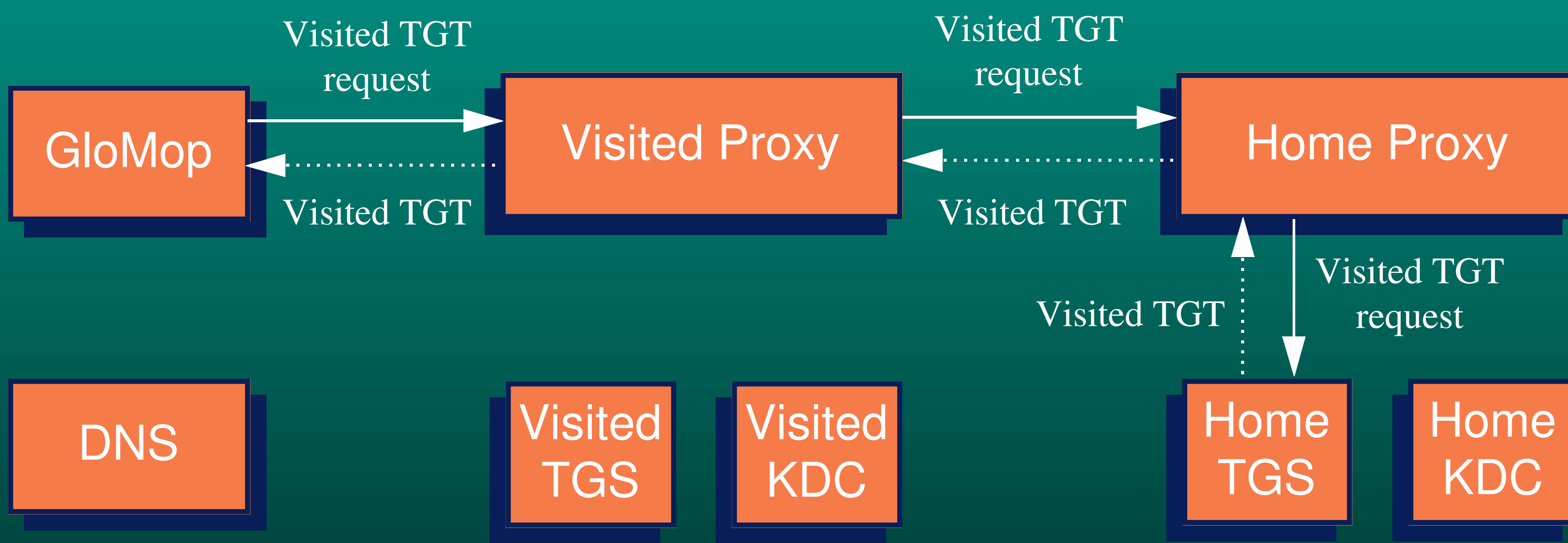
Authenticated Proxied Services

- proxies are treated as principles
- clients access proxies by gaining tickets for their services
- cross-domain authentication is possible by having a foreign TGS listed as a principle in the local system
- Charon is a protocol that enables Kerberos for impoverished clients
- Charon also enables “indirect authentication”

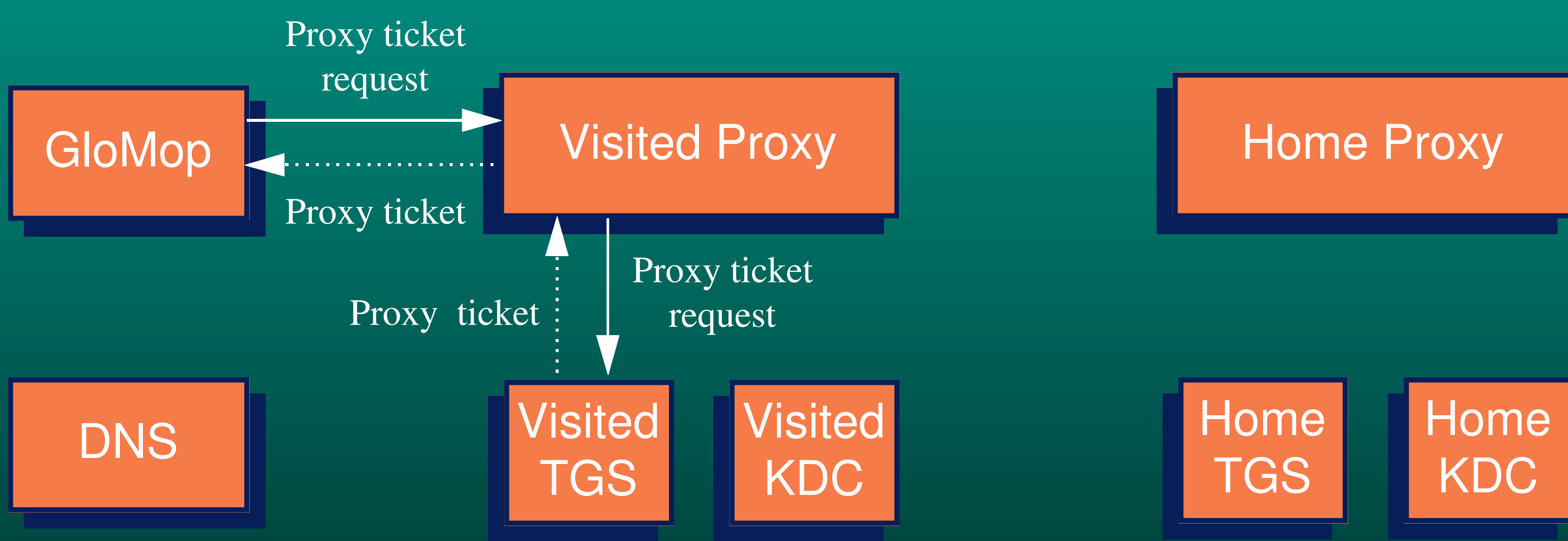
Charon Authentication: Obtaining TGT for Visited Proxy



Charon Authentication:
Obtaining TGT for Visited Proxy

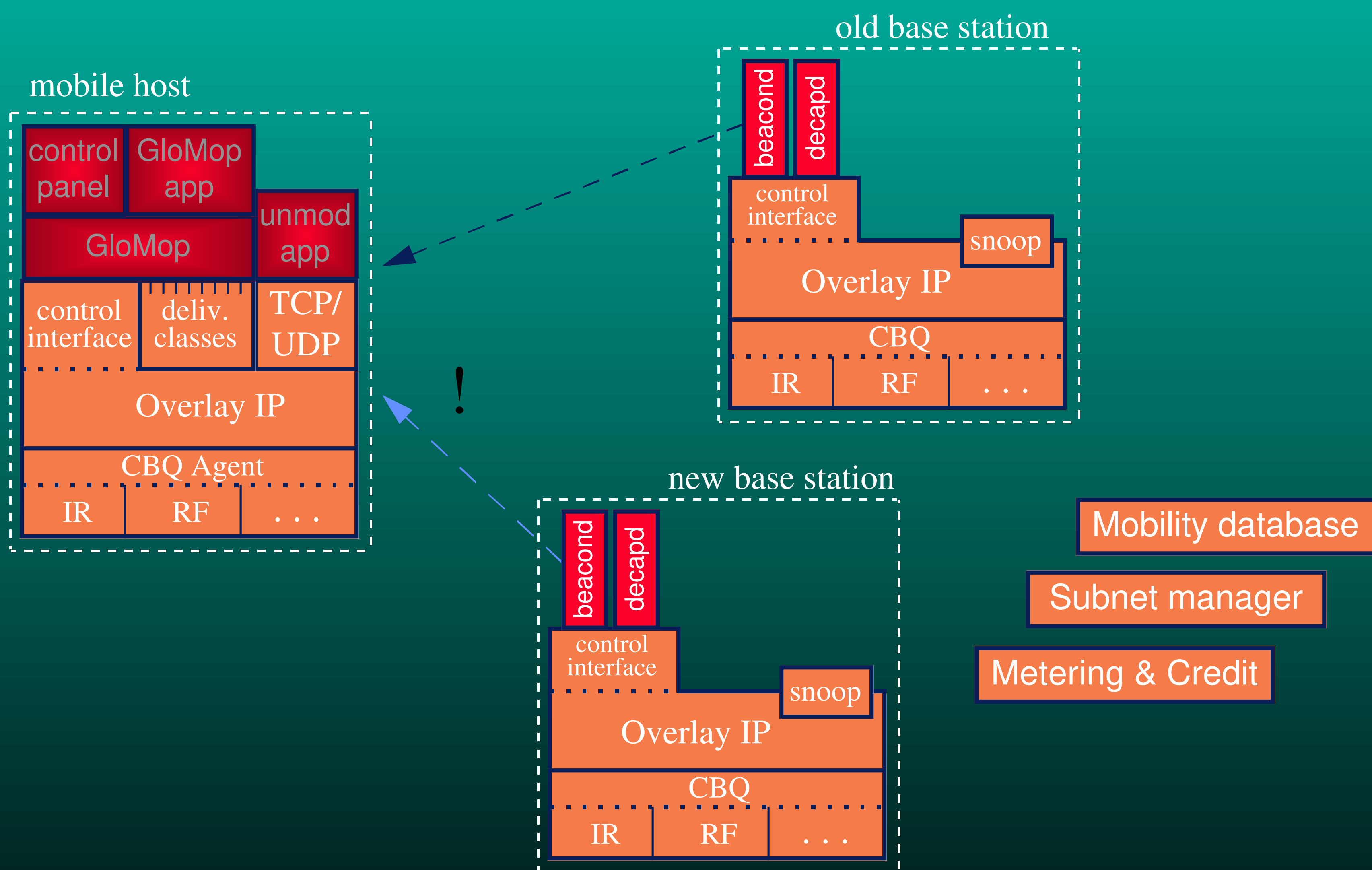


Charon Authentication: Obtaining Session Key with Visited Proxy



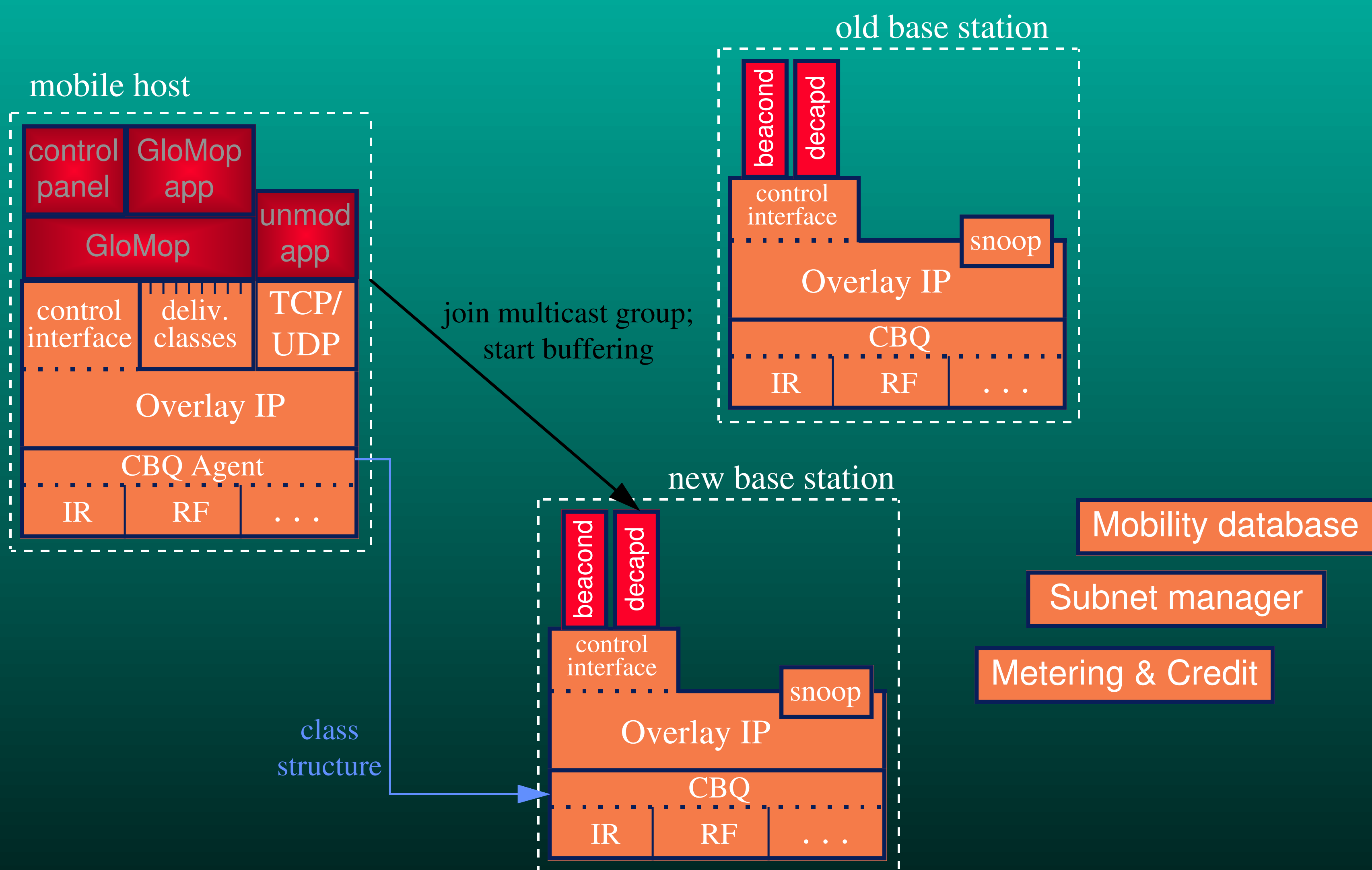
Handoff from IR to WaveLAN

(load balancing)



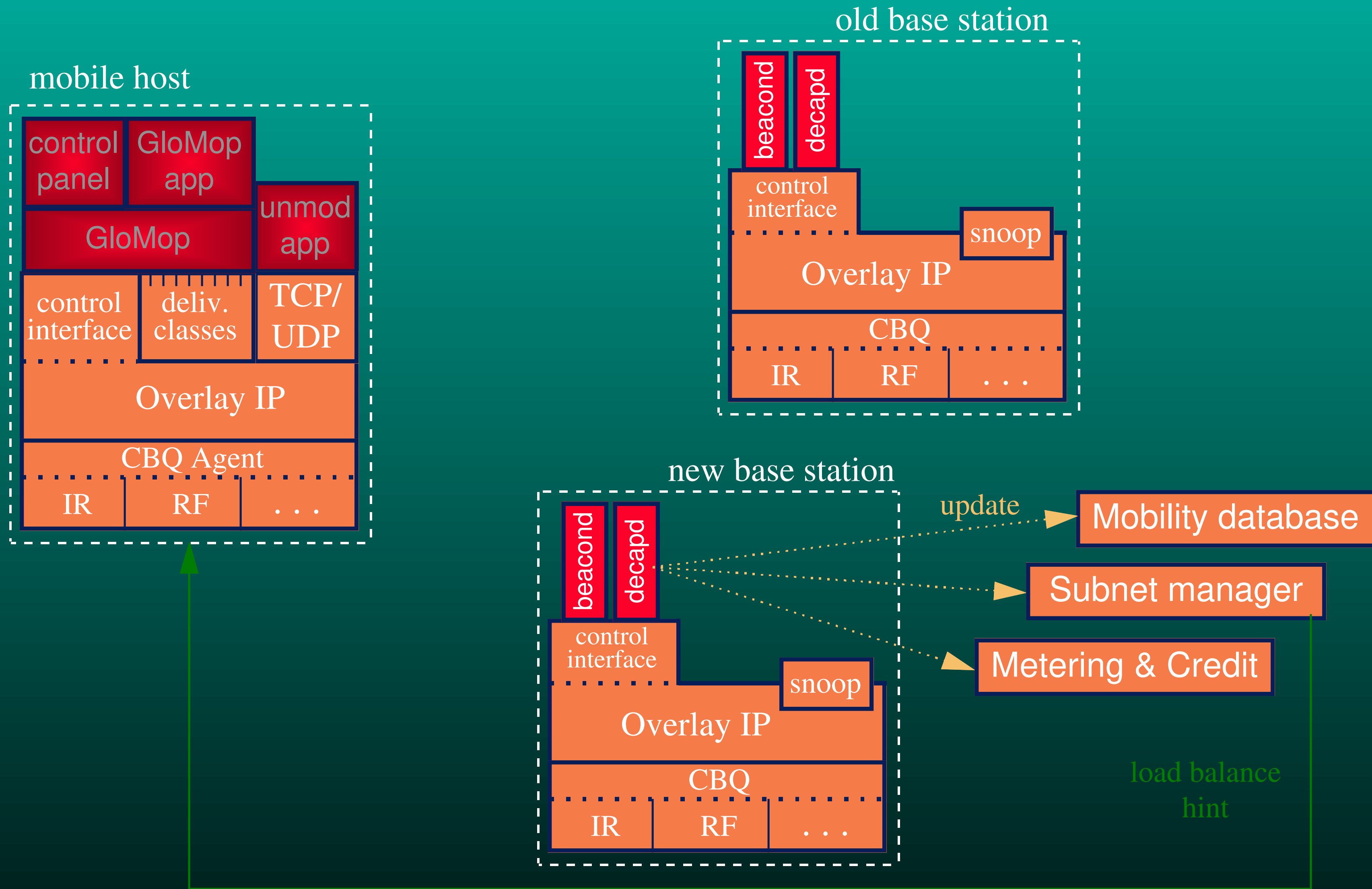
Handoff from IR to WaveLAN

(load balancing)

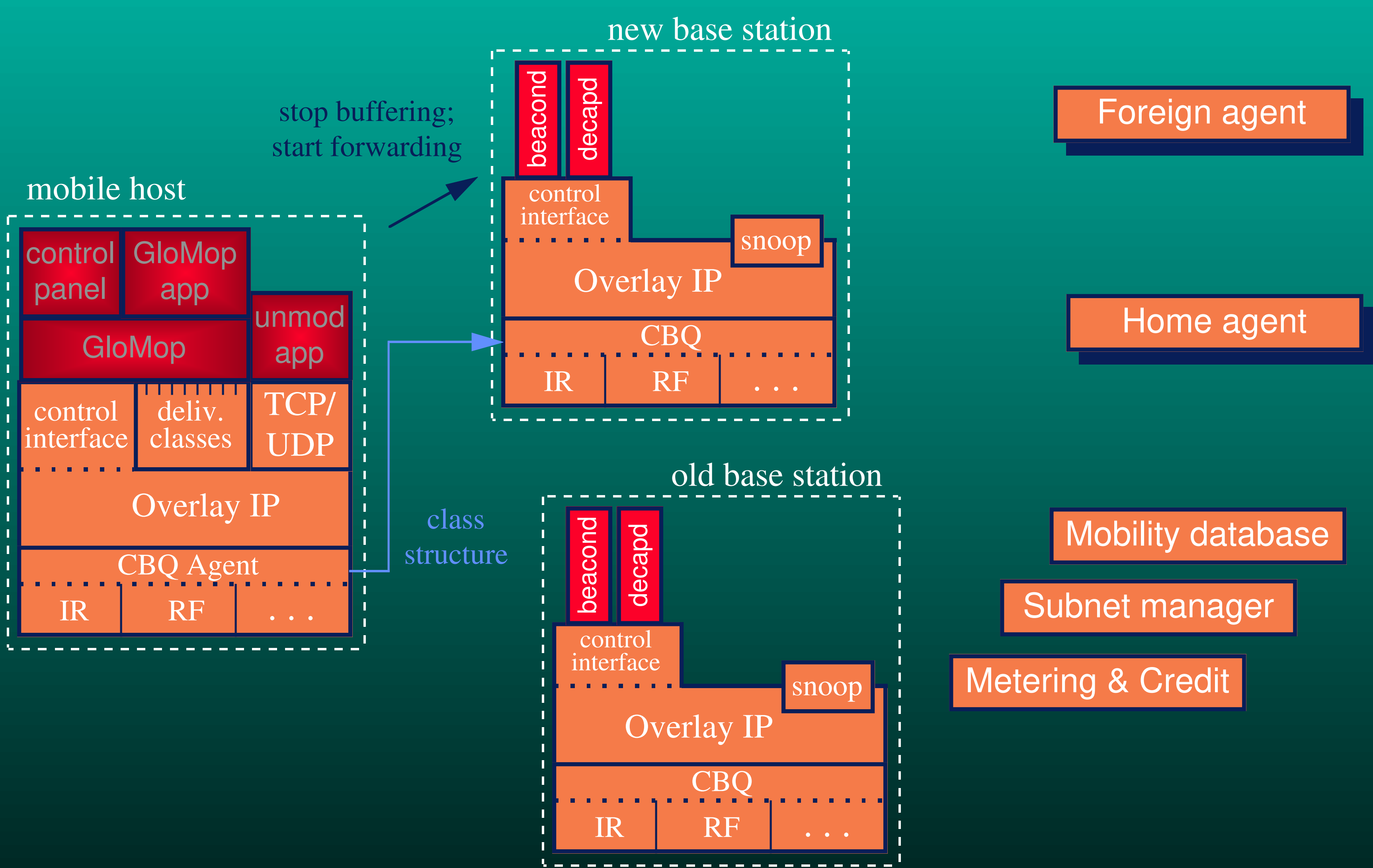


Handoff from IR to WaveLAN

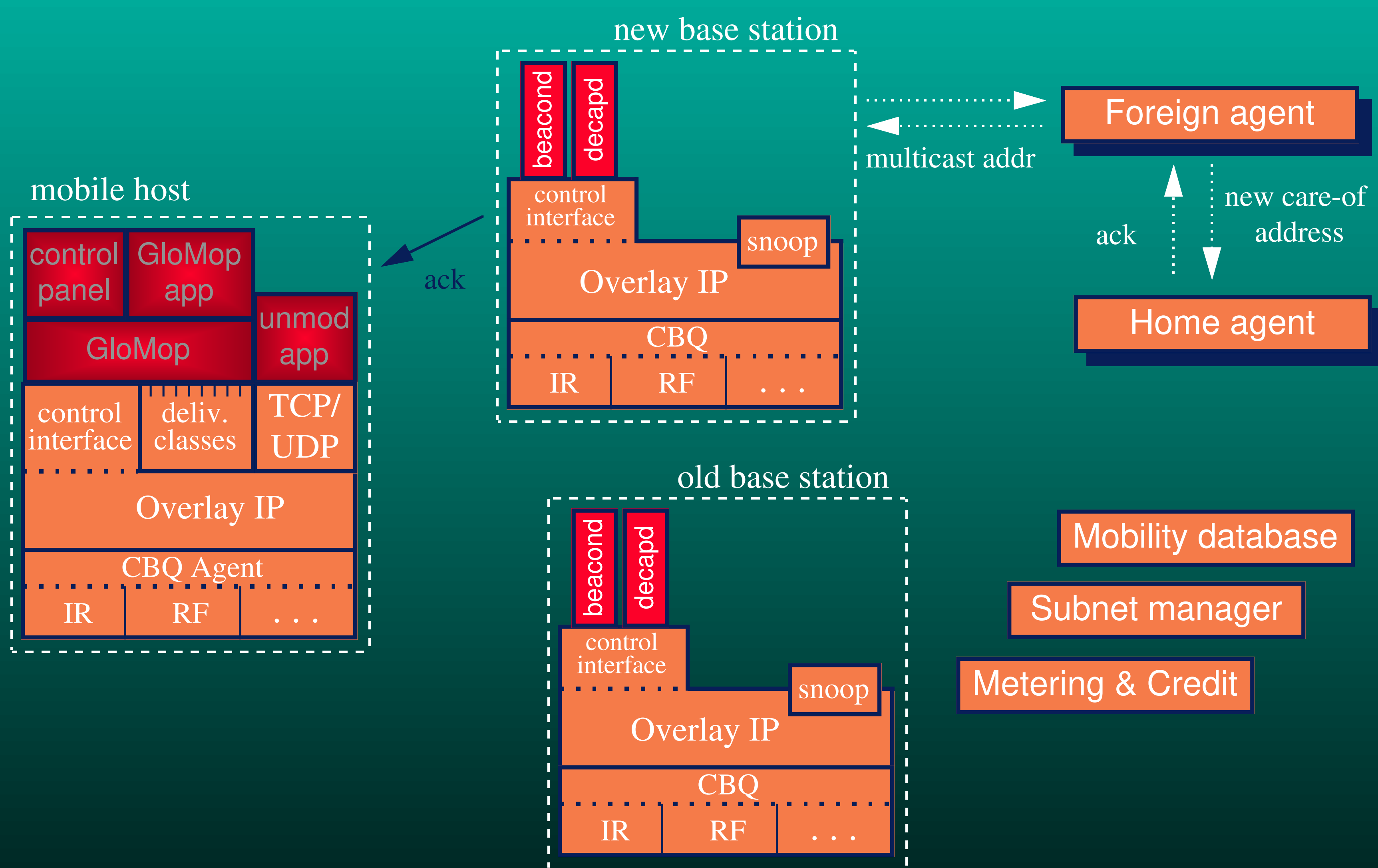
(load balancing)



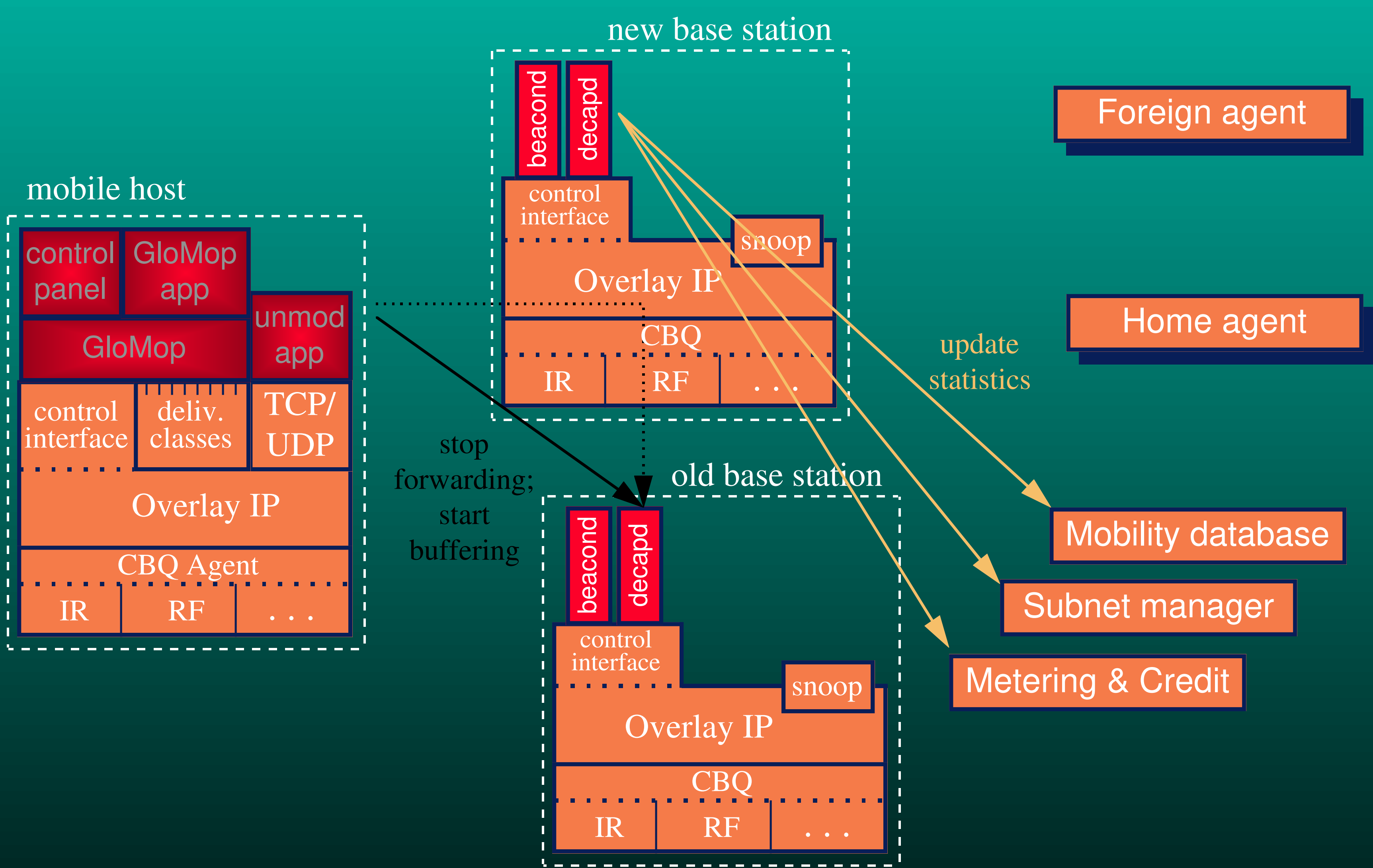
Extended Handoff Subroutine



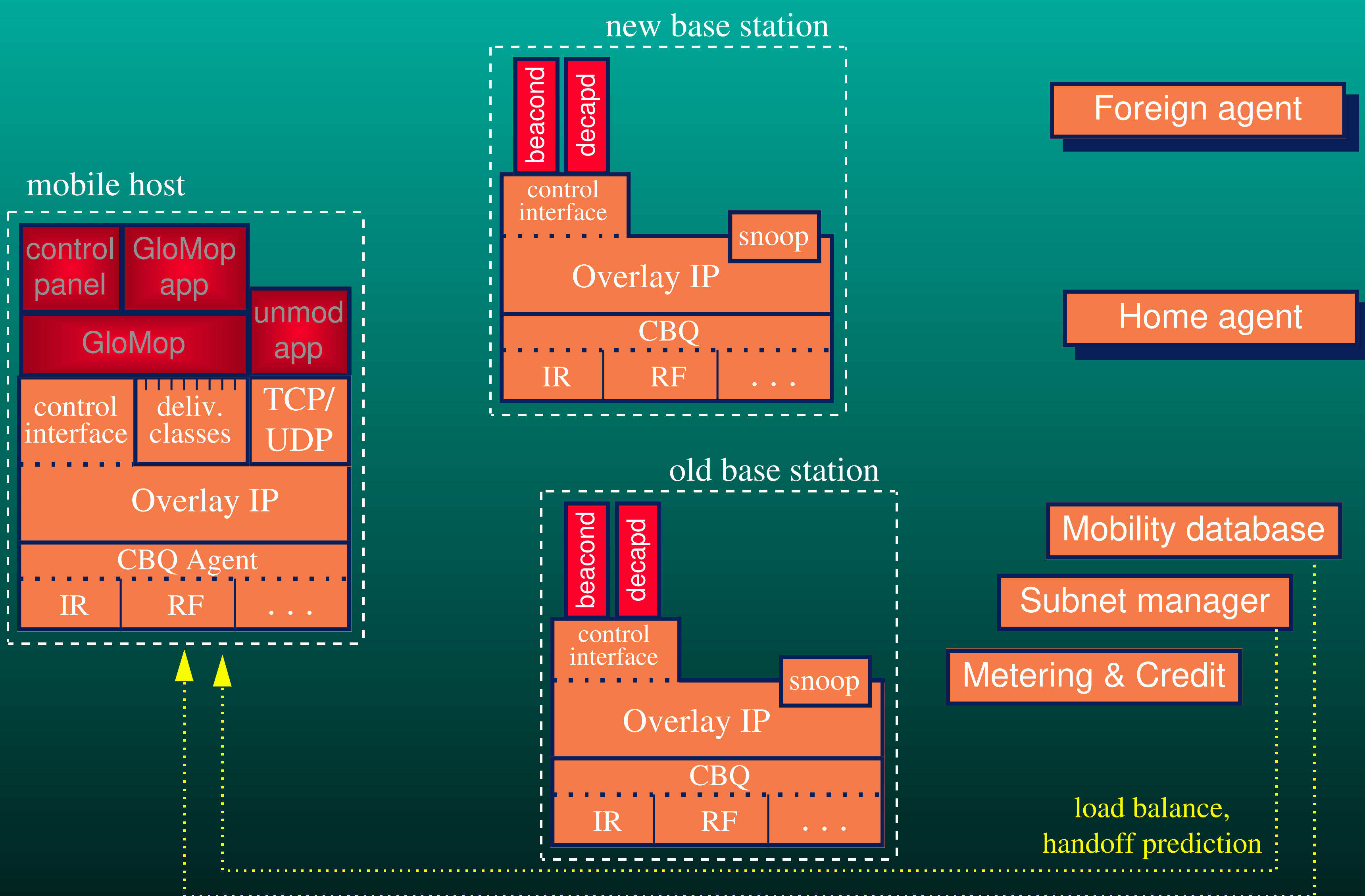
Extended Handoff Subroutine



Extended Handoff Subroutine

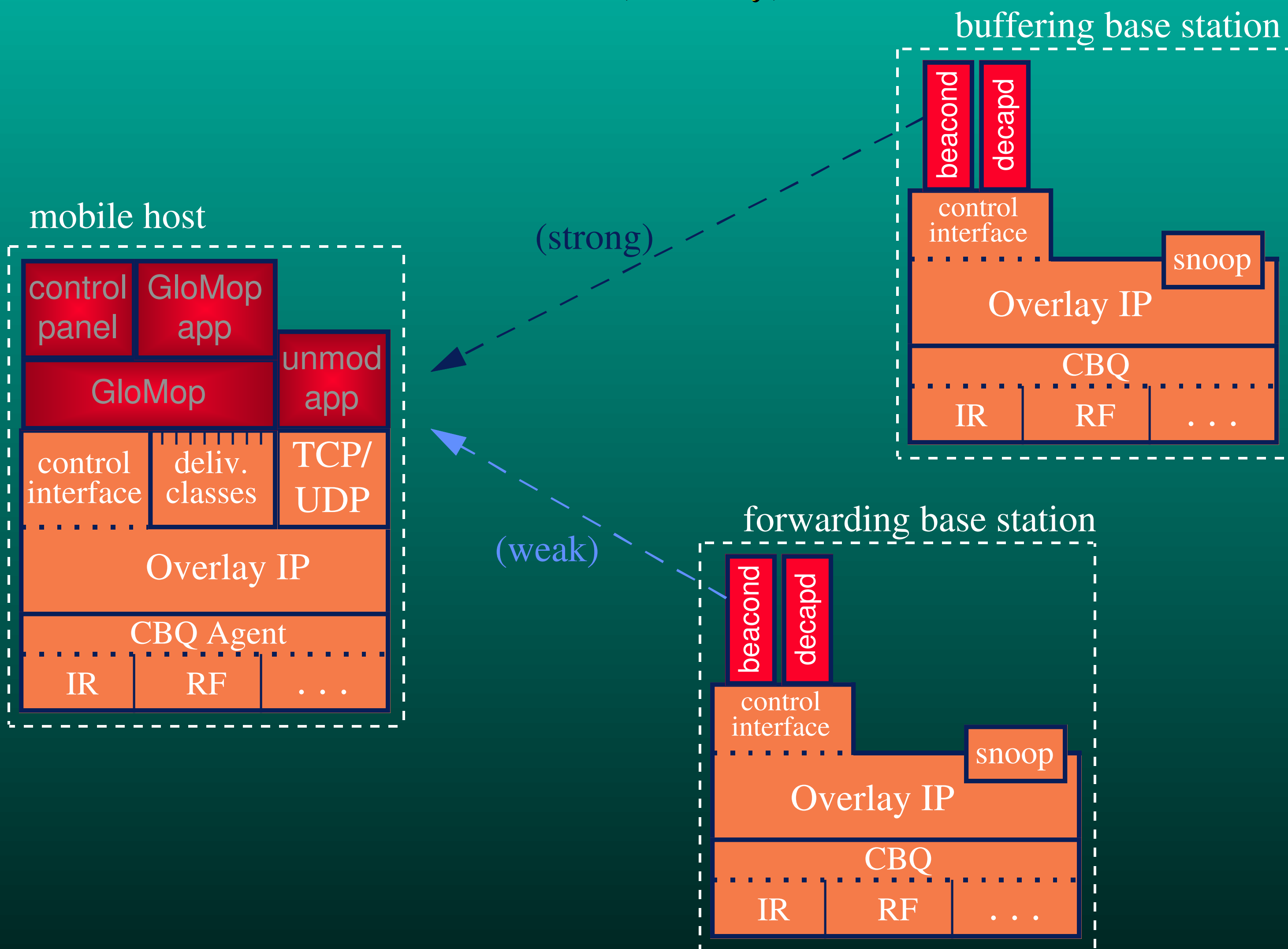


Extended Handoff Subroutine

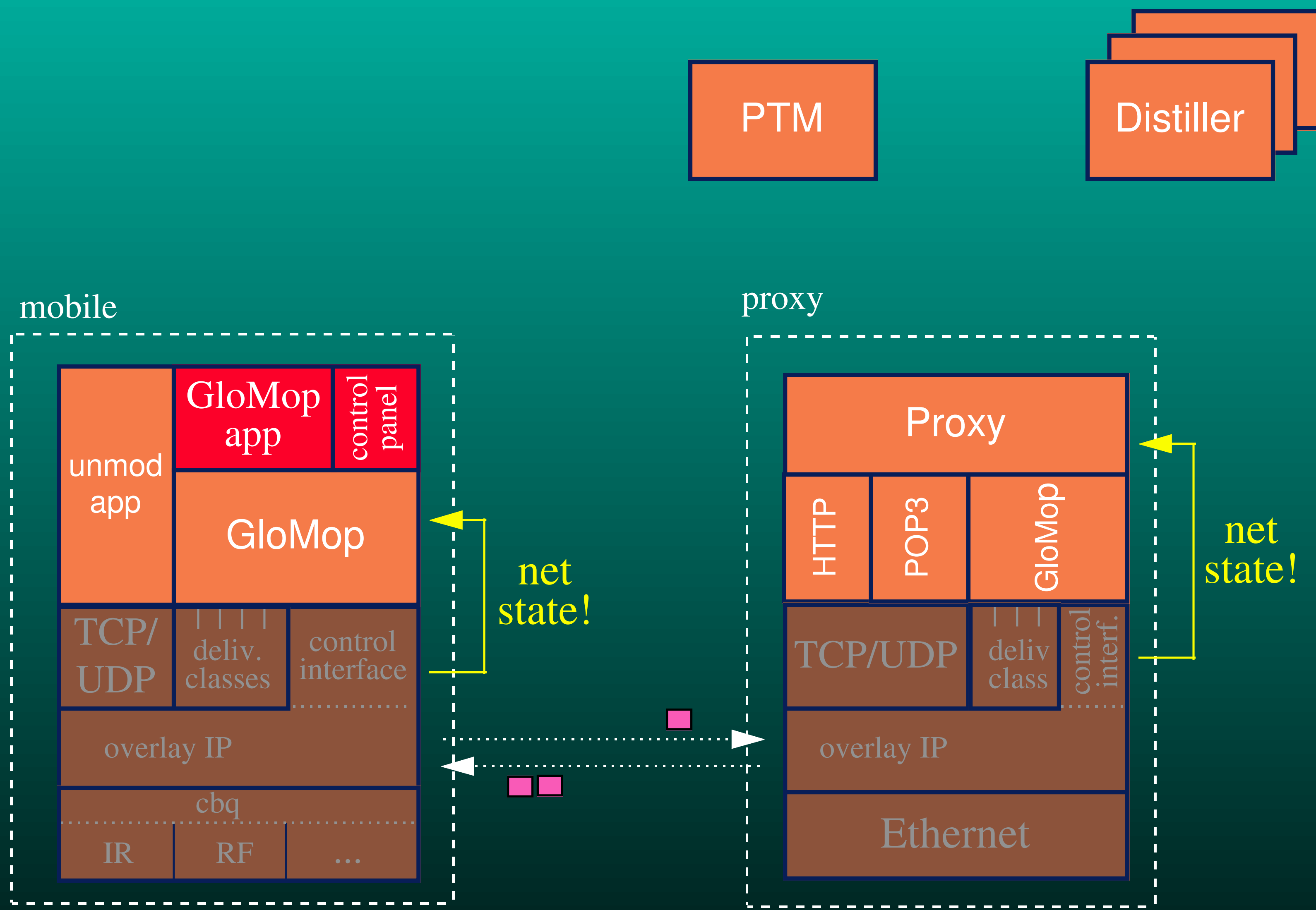


Handoff from WaveLAN to WaveLAN

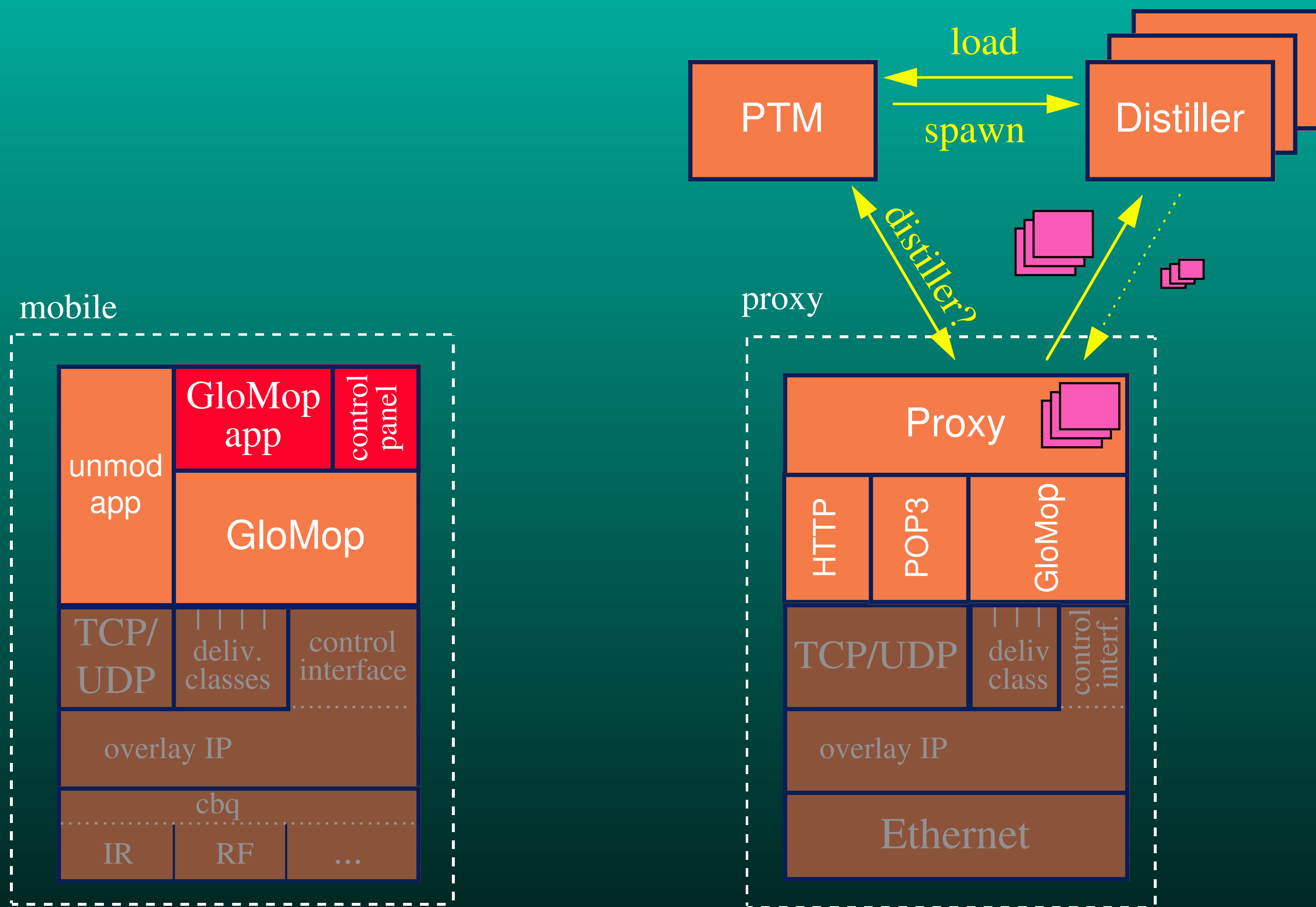
(mobility)



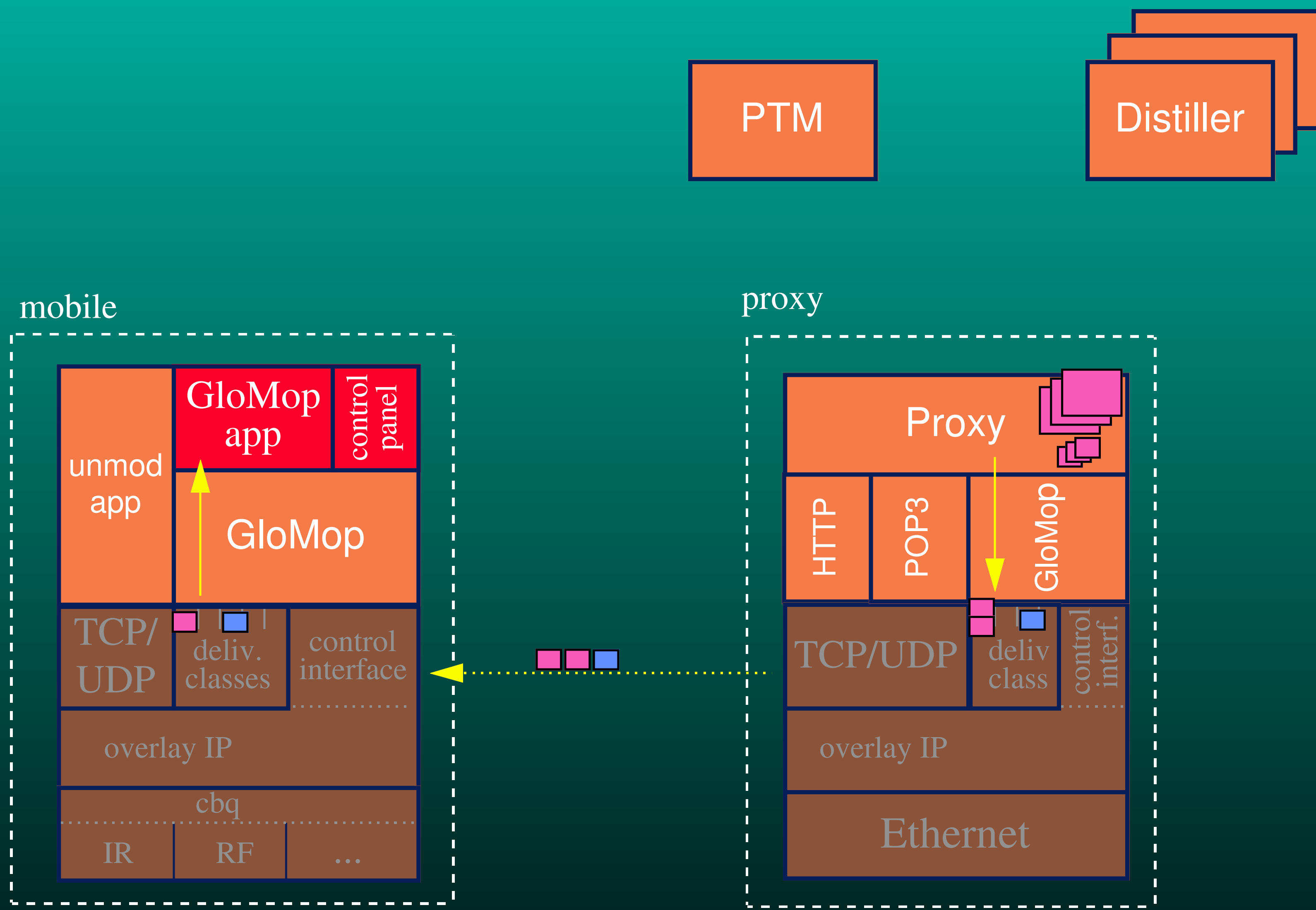
Dynamic Adaptation via NCM



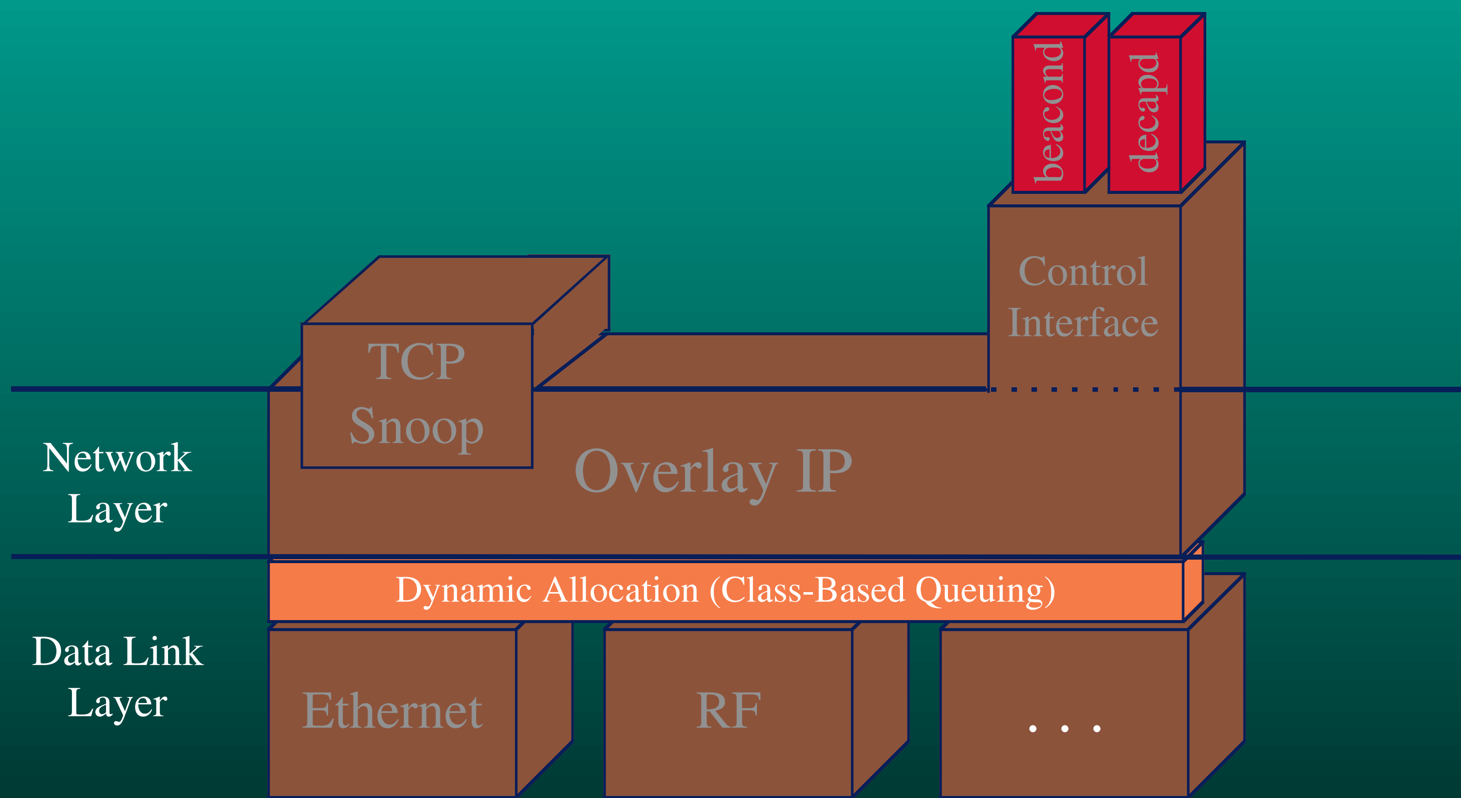
PTM Load Balancing



Delivery class abstractions

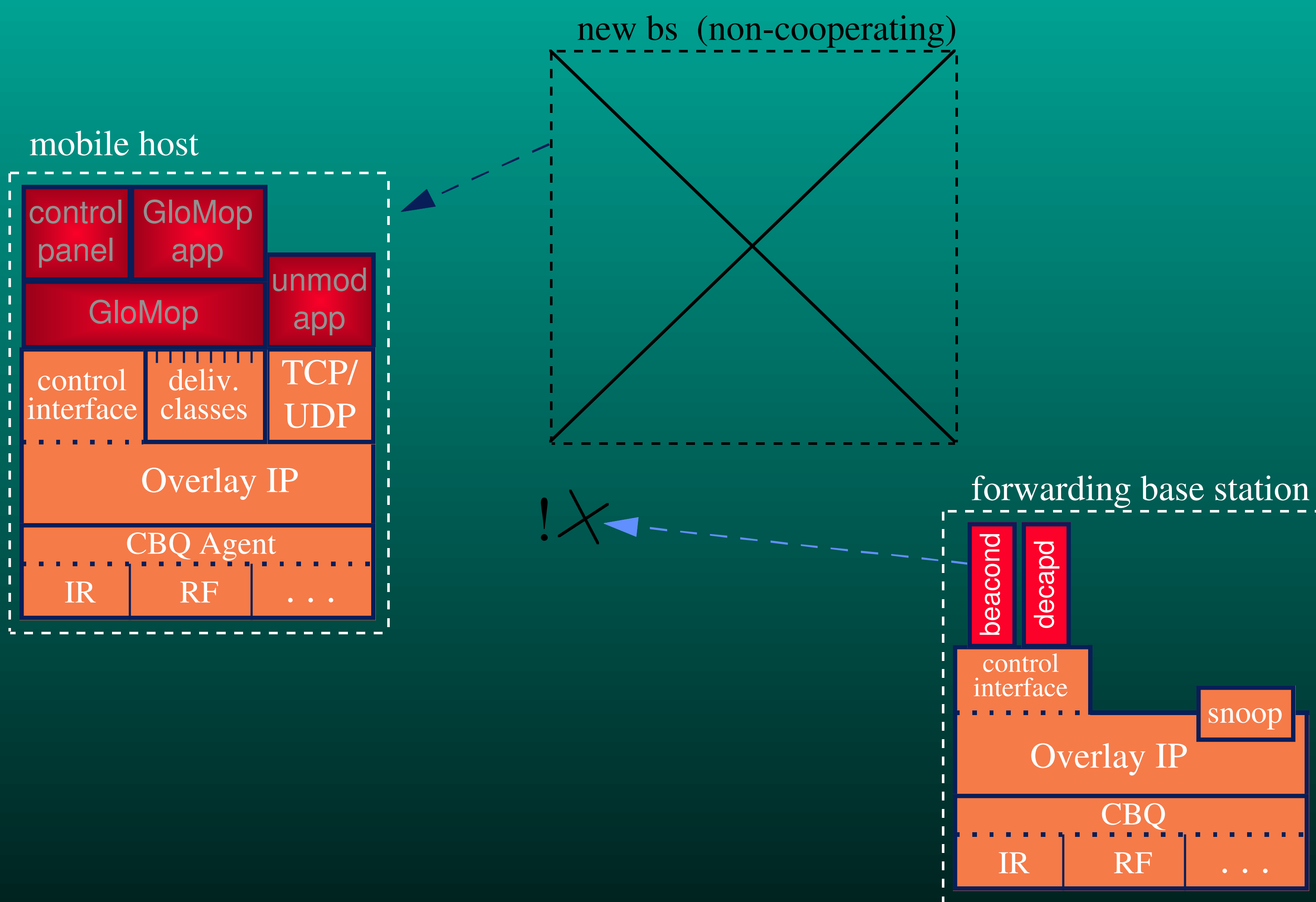


Dynamic Link Allocation

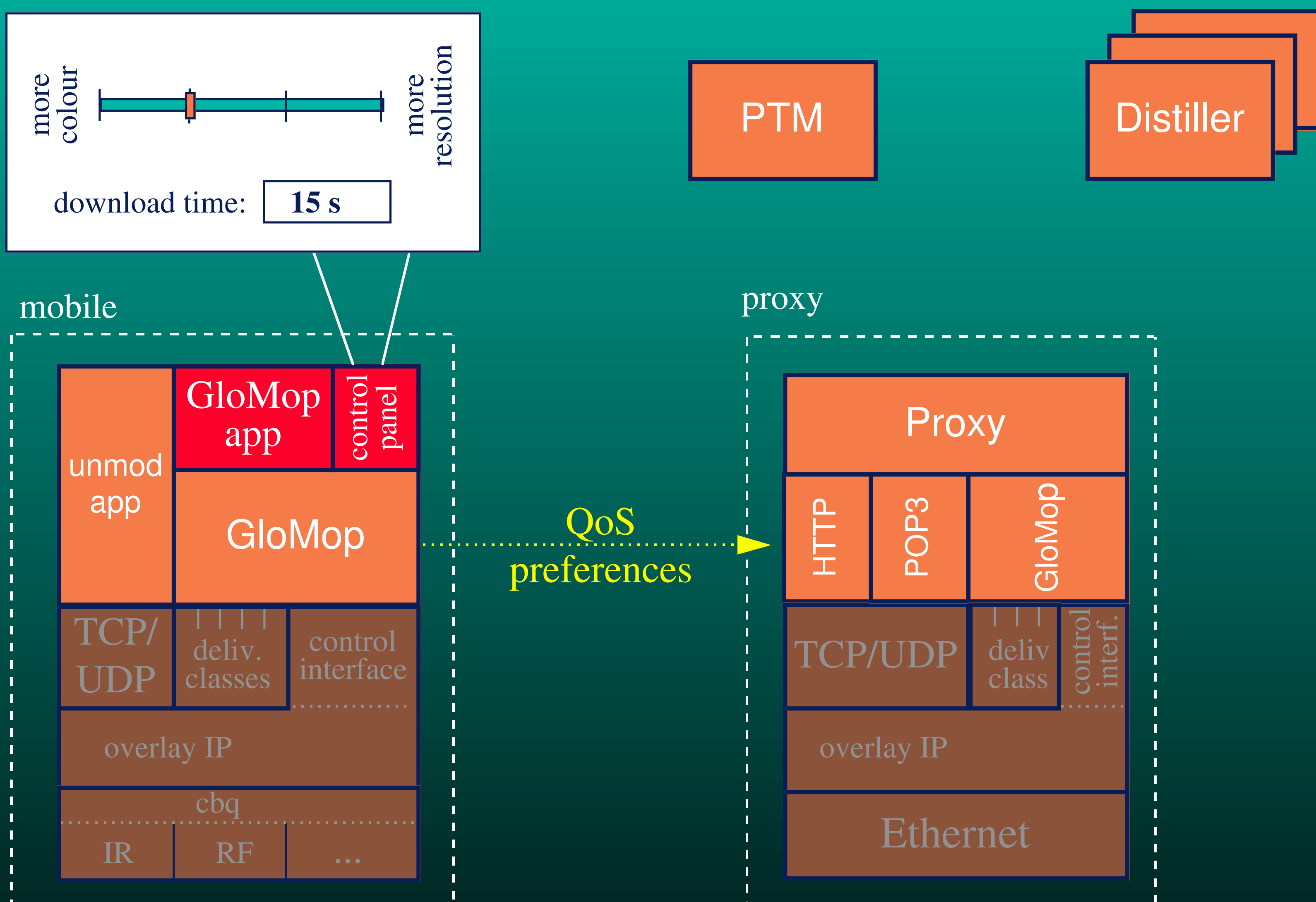


Handoff from WaveLAN to Metricom

(lost beacons)

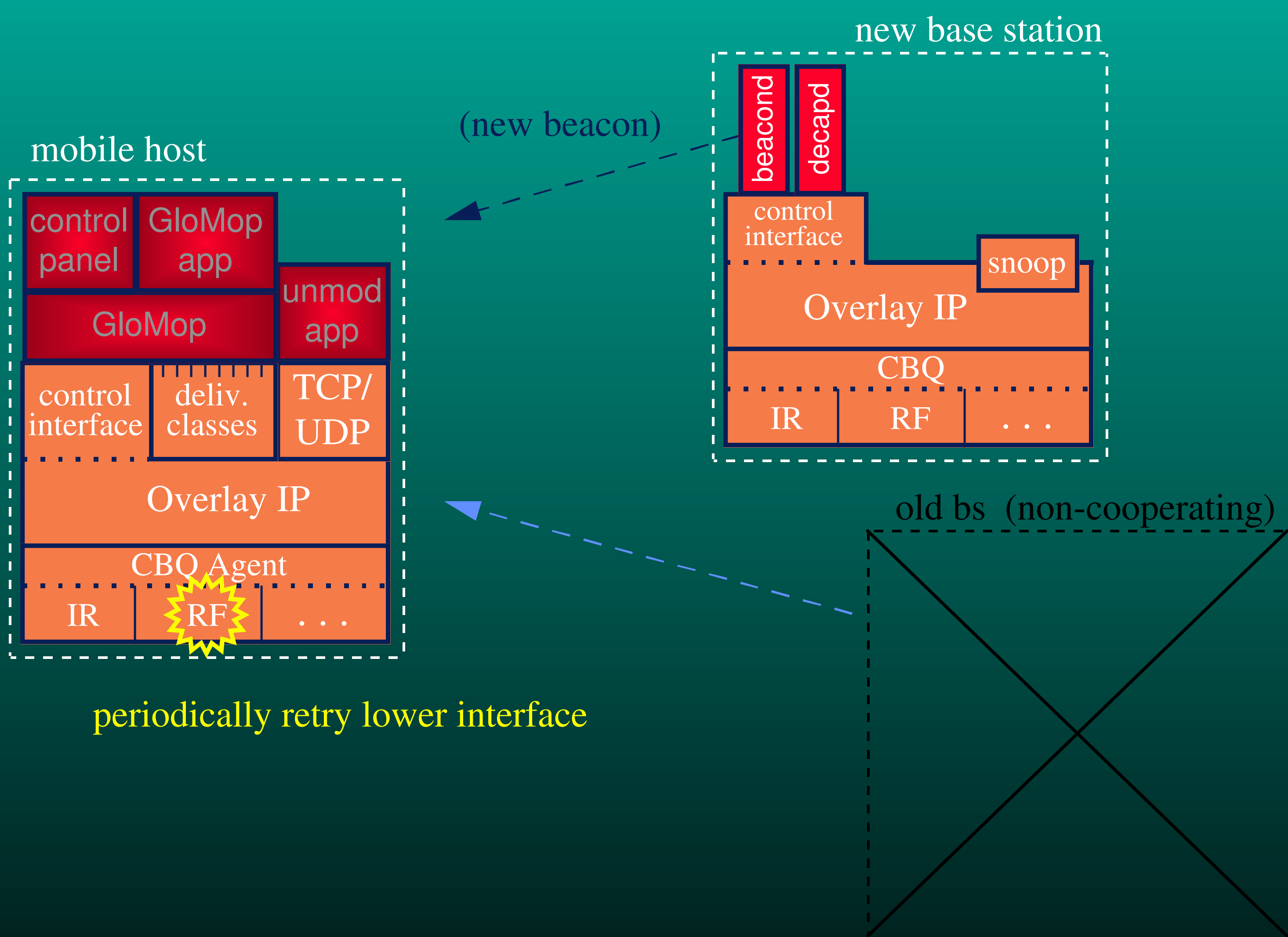


Flexible Refinement/QoS Mechanism



Handoff from Metricom to WaveLAN

(change of domain)



periodically retry lower interface

Outline

- Motivation and overview
- Core architecture components
- Core architecture walkthrough
  - connection, document retrieval, vertical handoff, document refinement
- Extended architecture components
- Extended architecture walkthrough
  - connection and authentication, vertical handoff, horizontal handoff, dynamic adaptation, load balancing, domain and network services
- Summary, Open Issues

Core Daedalus Architecture Summary

- Supports seamless roaming in heterogeneous and foreign environments
  - vertical and horizontal handoff support
- Custom Network Stack at MHs and (cooperating) BSs
  - unmodified corresponding hosts, home agents
  - compatibility with Mobile-IP
- Static assumptions about network characteristics
- Accepts handoff “hints” to handoff controller
- Some protocol-specific network optimization
  - TCP snoop

Extended Daedalus Architecture Summary

- Adds custom network stack at proxy host
  - delivery class abstraction for data-type specific transport
- Dynamic allocation at data link (hop by hop)
- Extensible, independent network services model
  - mechanisms facilitate localized policy decisions
- Dynamic measurement and notification of changes in network characteristics (via NCM)
- Network performance enhancements
  - low-latency handoff
  - hierarchical foreign agents
  - additional transport-layer optimizations

Core Proxy Architecture Summary

- Unmodified apps, application-specific protocols
  - proxy mechanism is either transparent to or already supported by the application
  - functionality and efficiency limited by the application’s protocol
- On-the-fly distillation and refinement possible
  - client and network adaptation, unmodified servers
  - proxy optimizations (prefetching, caching)
  - dynamic network adaptation is missing (no NCM)
- Loose coupling with Daedalus stack

Extended Proxy Architecture Summary

- Pluggable proxy architecture
  - support for both modified and unmodified apps
  - separation of load-balancing concerns into PTM
- GloMop application support layer
  - efficient custom protocol
  - explicit refinement and QoS mechanisms
- Authenticated proxied services
- Tight coupling with Daedalus stack
  - dynamic network adaptation via NCM
  - allow delivery policy through delivery classes

Some Open Issues

- Elements of the service architecture
  - complex element interaction and inter-dependency
  - the metering mechanism is not fully resolved
- Multiple simultaneous network interfaces
  - multiplexing of application data
  - undesirable for power-management
- Link-layer state on handoff
  - is state transfer necessary?
  - how to deliver data queued in base-station
  - different link-management policy in new network?
- Network connection monitor
  - stability and granularity of statistics

Some Open Issues (continued)

- Function migration
  - "Evil Twin" versus protocol filters at the proxy
  - can Rover-style migration coexist?
- Time-constant for proxy-adaptation
  - how quickly does it adapt, and to what granularity of network variation?
  - should the proxy receive adaptation hints?
- User interface issues
  - understanding and specifying constraints
  - data-type specific refinement controls
  - per-chunk, per-document, and per-session constraints