

# PA2 common questions and SMTP protocol

# Common questions

- When to perform triggered updates? Include first entry?
- How to iterate through entries in routing packet?
- How to purge an entry?
- Routing heuristics

# When to perform triggered updates?

- According to spec, “a vrouter sends a *triggered update* whenever there is an increase in any of its entries' metric. A link going down also causes the propagation of a triggered update”
- If unspecified, don't do triggered update

# When to send first entry

- First entry = vrouter's own address
  - Cost to itself is always 0
- For periodic update,
  - “A vrouter's entire routing table is sent to all its neighbors every VRP\_UPDTSECS.”
- For triggered update, by previous definition
  - An entry will be sent in the update only when the link cost has gone up or the link is going down (why not do triggered update when cost goes down)
  - None of the two satisfies, so never send it in this case

# How to iterate through entries in routing packet?

```
struct _vrp {  
    u_short vrp_ver;      /* vrp version */  
    u_short vrp_type;     /* VRP_TYPE{DV,PV,LS} */  
    vin_addr_t vrp_dv;    /* first of an array of distance vectors,  
                           vad metric is cost to destination */  
};
```

- When allocating the memory for the routing packet  

```
vrp_t *vrp = malloc (size);          //size is computed based on how many  
                                     //entries you have to propagate.
```

  
    NOT 

```
vrp_t *vrp = malloc(sizeof(vrp_t));
```
- When you receive such packet:  

```
for(vin_addr_t* i = &(vrp->vrp_dv); i < num; i++)  
{  
    //i is the routing entry  
}
```

# How to purge an entry?

- Simple way is to memset the entry to 0
- Or set each individual field of an entry to 0 or NULL
- Or have you own entry management scheme
  - Whenever an entry is purged, move the rest of the entries up
  - Can always add a new entry at the end

## -o flag

- In command line, specify -o will produce the routing table entry to help you debug

# Routing heuristics – **Triggered update**

- Triggered updates are an attempt to speed up convergence.
- Don't reset the timer for periodic update

## Routing heuristics – **Route poisoning**

- Advertise cost as infinity when cost from next hop has been increasing and continue to stay with current next hop for "a while".

# Routing heuristics – **Split-horizon with poison reverse**

- It is never useful to claim reachability for a destination network to the neighbor(s) from which the route was learned.
- "Split horizon" is a scheme for avoiding problems caused by including routes in updates sent to the node from which they were learned.
- The "simple split horizon" scheme omits routes learned from one neighbor in updates sent to that neighbor.
- "Split horizon with poisoned reverse" includes such routes in updates, but sets their metrics to infinity.

## Routing heuristics – **Path hold-down**

- Do not switch route for n update periods after cost goes up. This reduces the likelihood of switching to a route that is invalid due to the count to infinity problem.
- n is defined in vrp.h as VRP\_HOLDNRND

# SMTP protocol

- Architecture
  - Mail sender -> Sender's mail server -> Receiver's mail server -> Mail receiver
- Performance
  - Pipelining
  - Not implemented correctly by many vendors (hotmail, yahoo mail)
- Security
  - Denial of Service attack (aggressive pipelining)
  - spoof sender email address via "mail from" command