Policy Control

in

RSVP

Shai Herzog
USC/
Information Sciences Institute

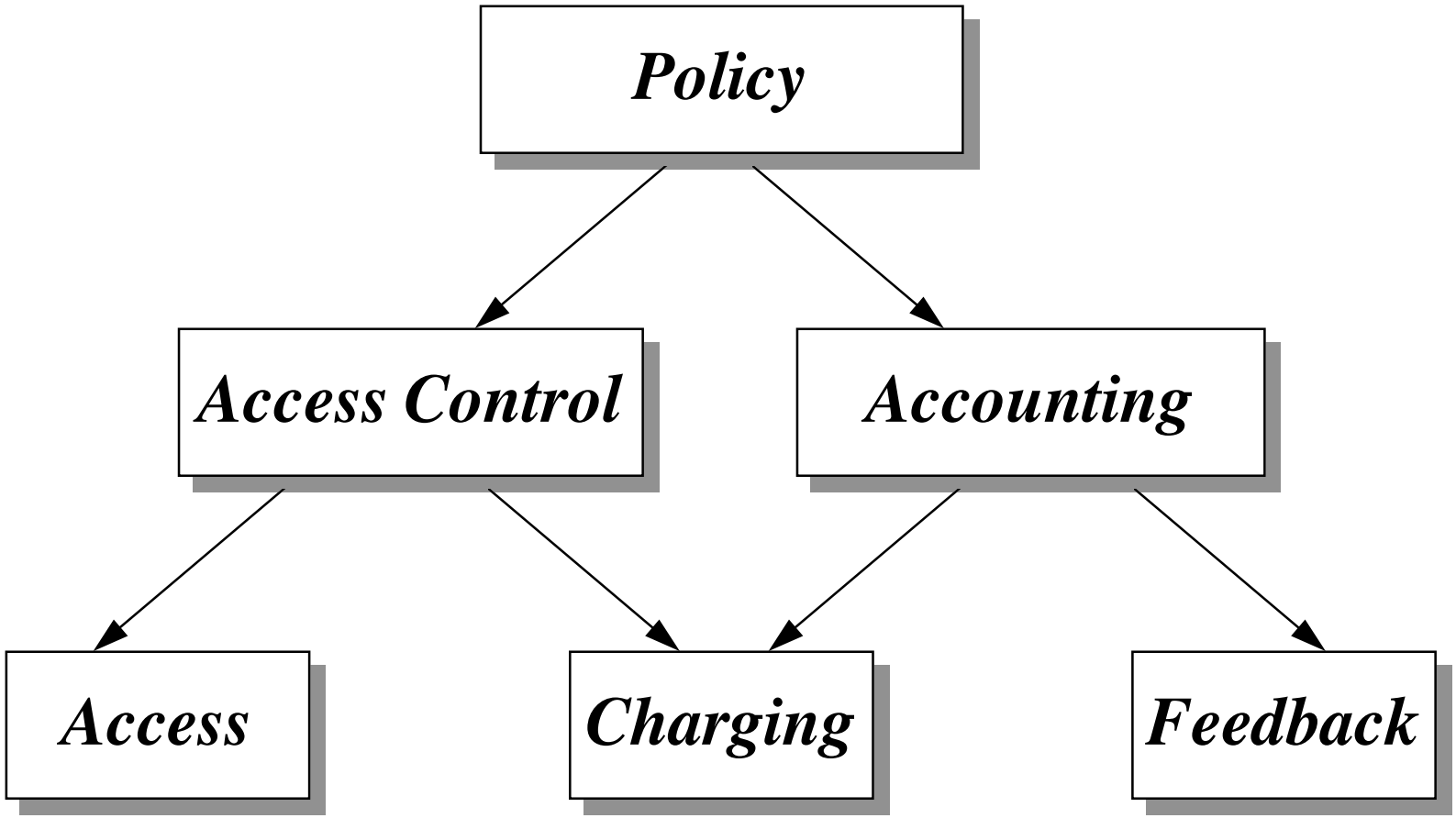
ftp://ftp.isi.edu/rsvp/draft-ietf-rsvp-policy-XXXX-00.{ps,txt}
XXXX = {arch, lpm, ext}

Document Breakup

- **Motivation**
  - Break into simpler, cleaner basic components
  - Allow parallel efforts by WG
- **Documents**
  - *Models/Architecture* (**-arch-**)
    - High level policy modeling and examples
  - *Mechanism/LPM* (**-lpm-**)
    - Recommended mechanism
    - Provides overview, services description, internal formats design and processing rules
  - *RSVP extensions* (**-ext-**)
    - Policy components visible to RSVP
    - Interface/processing rules
    - I-D May merge with the RSVP spec in the future

Models / Architecture

Policy Admission Control

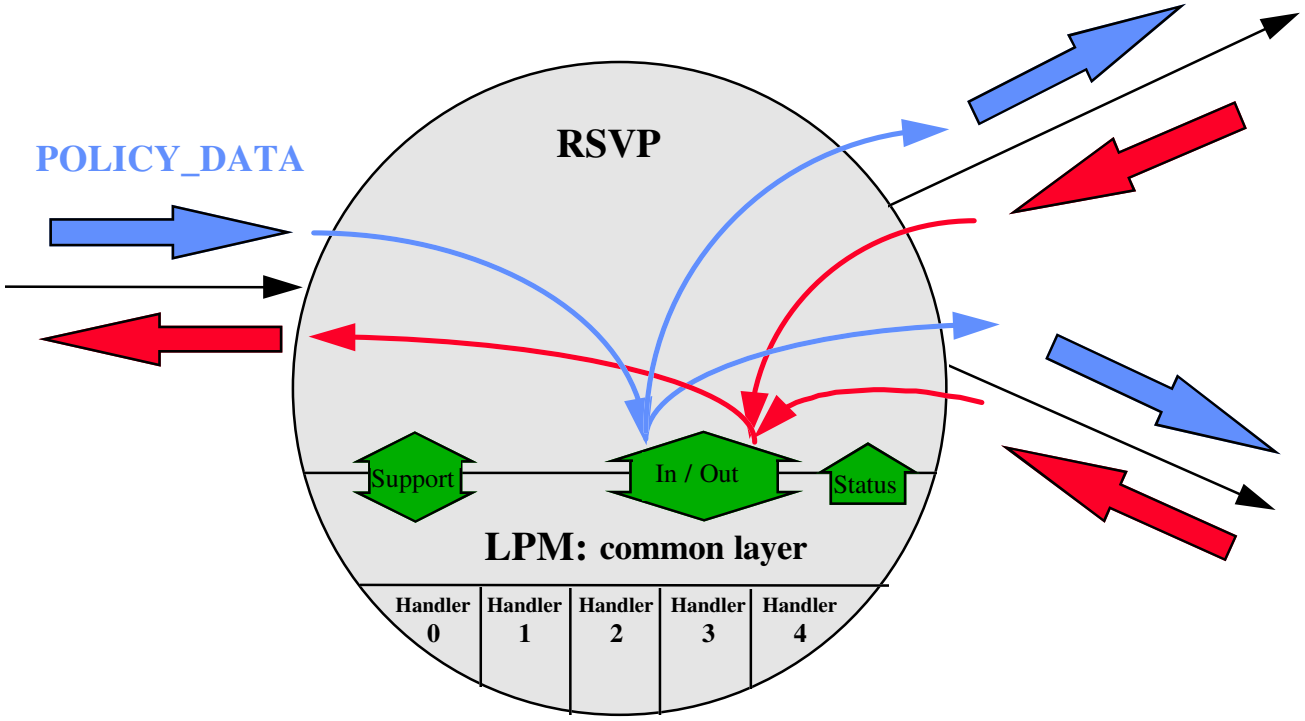


Models: Basic Principles

- **Described in March & Dec. meetings...**
- **Bilateral agreements**
  - Between neighboring service providers/consumers
  - Motivated by scaling consideration
- **Policies are controlled/configured locally**
  - Could be based on
    - Bilateral agreements
    - Reservation related Information (credentials etc...)
    - Local policies

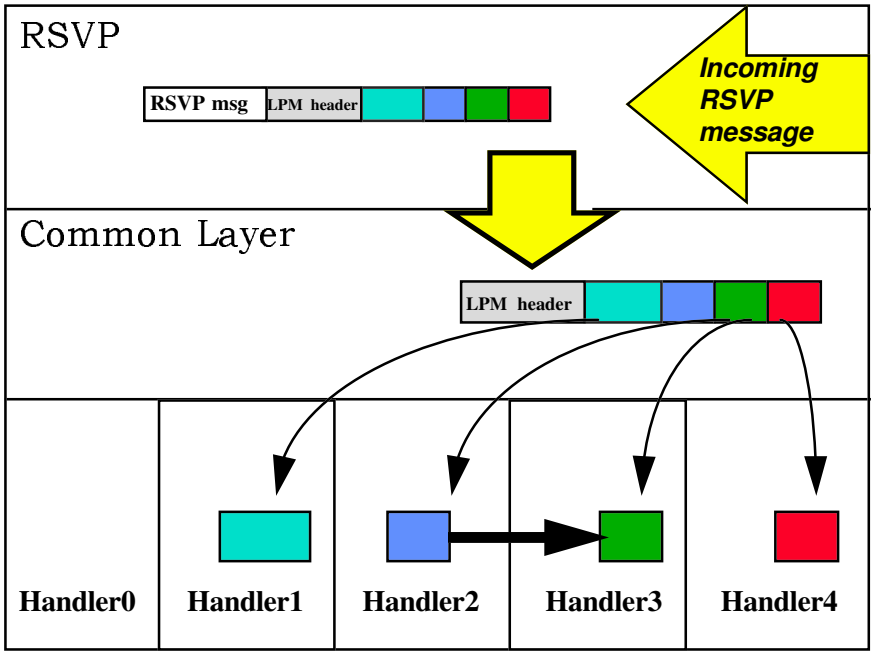
Mechanism: LPM

Mechanism: LPM



Local Policy Modules (LPM)

- **Separate Box from RSVP**
  - Separate internal state
  - Linked through handles
- **Hooks to RSVP**
  - POLICY_DATA objects
  - Piggy-backed on RSVP msgs
  - Define the set of LPM services
    - Model: “Policy based” adm. control
- **Policy configuration**
  - Handlers are the basic building blocks of policies. Ex:
    - Handler1: How much ?
    - Handler2: Who is debited ?
  - LPM implements policies by dynamically binding handlers

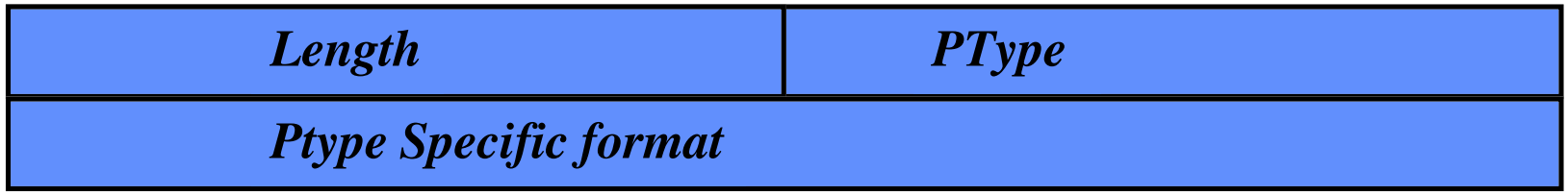


Generic LPM services

- ***lpm_open()*** — **Link RSVP to LPM state through fh's (flow handles)**
- ***lpm_close()*** — **Releases an fh and purges its LPM state**
- ***lpm_in()*** — **Syntactic processing of incoming POLICY_DATA obj.**
- ***lpm_out()*** — **Assemble outgoing POLICY_DATA objects**
- ***lpm_status()*** — **Verify the status of a reservation**
  Optional: debiting for reservations
- ***lpm_prune()*** — **Purge LPM state on a specific "branch"**
- ***lpm_config()*** — **Initialize the LPM policy from local configuration files**

Policy Attributes

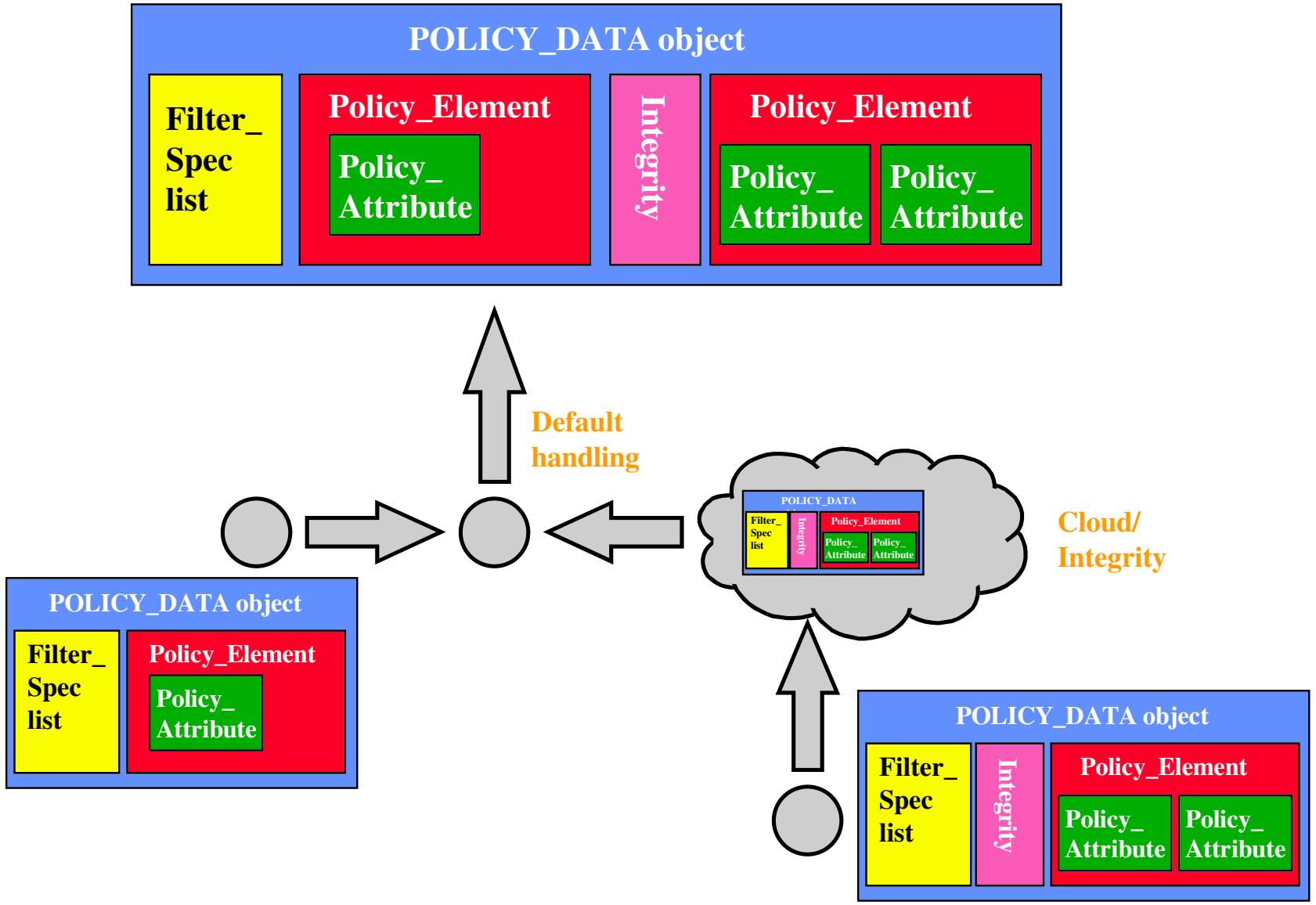
- **Format**



- **Specific formats**
  - Scoped (local) and unscoped policies
  - WG effort
    - We expect people to tell us what they need/want
    - Where should policies be listed?
      - Appendix to LPM document
      - Appendix to architecture document
      - Fourth document

RSVP Extensions

Policy Objects (Resv msg)



Semantic Fragmentation

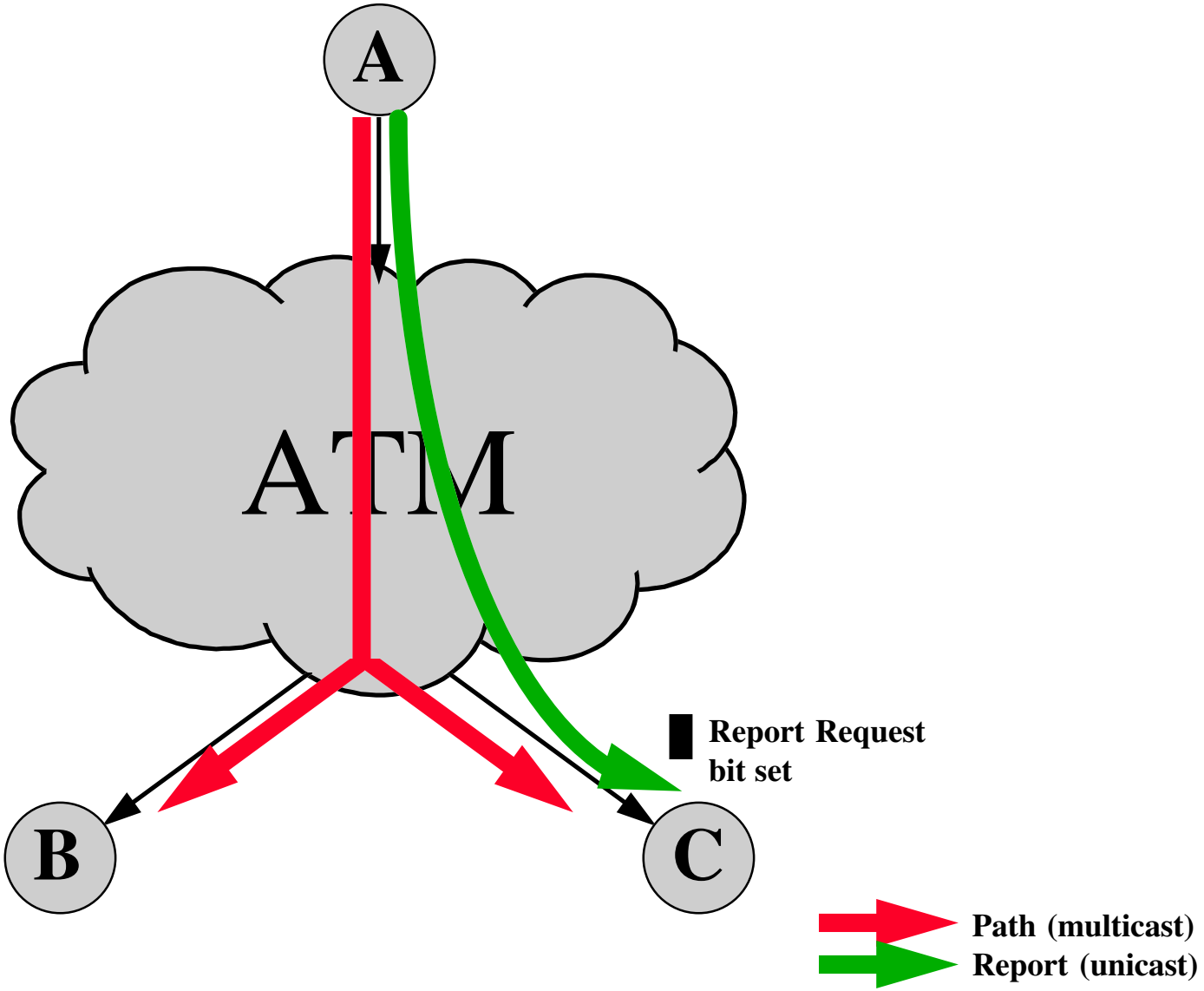
- Policies are additive in nature
- Fragment to POLICY_ELEMENT level
- State maintenance:
  - Individual timers for POLICY_ELEMENTs
  - LPM implementation issue
- Example

$[H, F, S_1, S_2, S_3, Pe_1, I_2, Pe_2, Pe_3]$

$\Rightarrow$ $[H, F, S_1, S_2, Pe_1, I_2, Pe_2, Pe_3]$
$[H, F, S_3, Pe_1, I_2, Pe_2, Pe_3]$

$\Rightarrow$ $[H, F, S_1, S_2, Pe_1, Pe_3]$
$[H, F, S_1, S_2, I_2, Pe_2]$
$[H, F, S_3, Pe_1, Pe_3]$
$[H, F, S_3, I_2, Pe_2]$

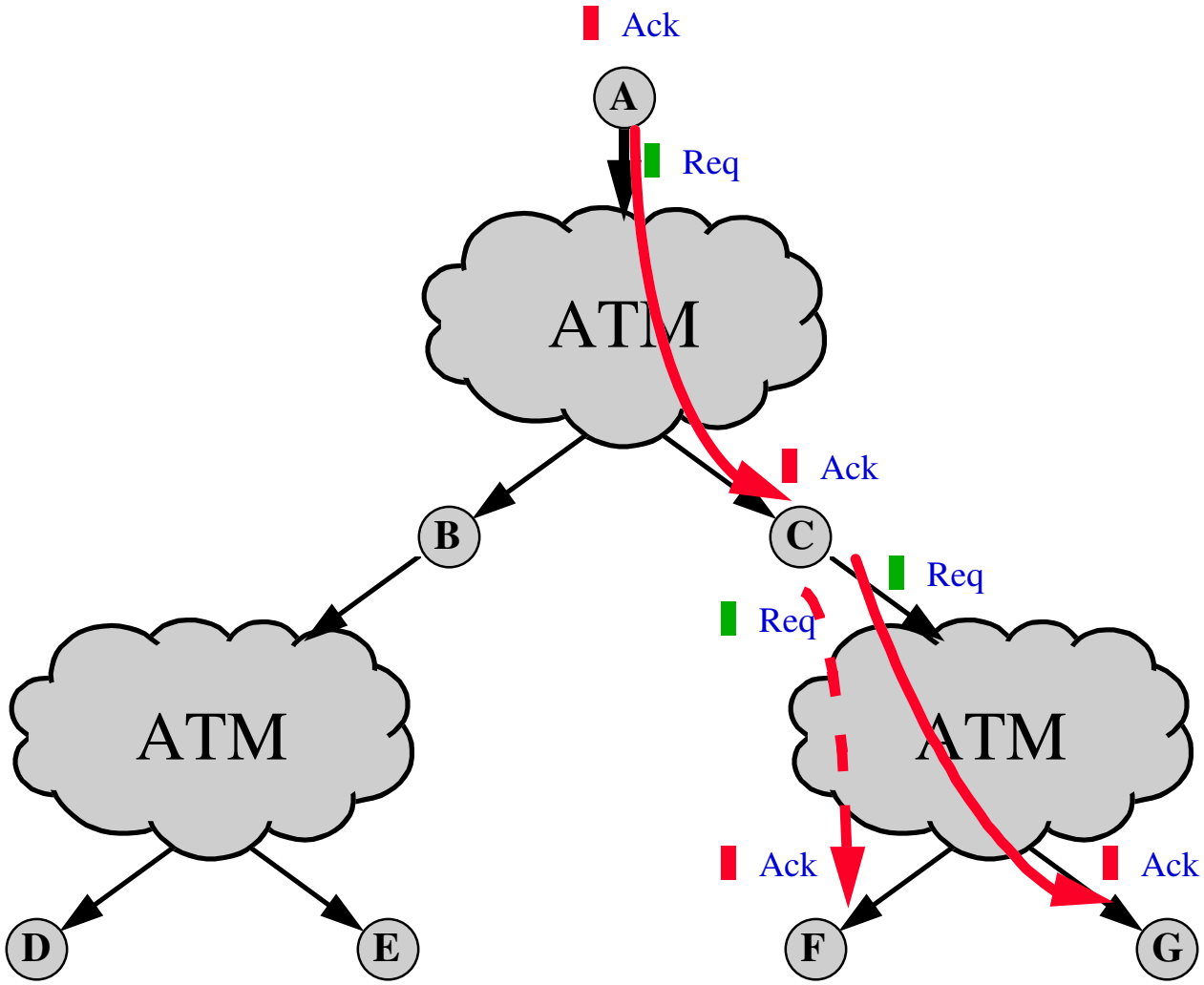
New RSVP message: *Resv Report*



Resv Report Message

- **Positive E-2-E policy response**
  - E.g.: Cost share of a receiver
- **Forwarding**
  - Periodic or triggered refresh
  - Unicast, Hop-by-hop, downstream, along reserved branches
  - Optional: Request bit
    - Delivered only to branches with bit set
    - Refreshes stop when bit is unset
- **Free ride...**
  - E-2-E ack: additional ack bit per incoming interface

E-2-E Acks over Resv Report



Policy Control Interface for RSVP

- **Separate POLICY from RSVP**
  - Do not pass FILTER_SPEC objects (use handles)
    - RSVP identifies its “flows” to policy module
      - Policy module shouldn’t know about resv. styles
      - RSVP associates source to receiver objects
  - Do not store POLICY_DATA objects in RSVP
  - Leave ***local*** policy processing out of RSVP (e.g. merging)
  - Support for state maintenance/synchronization
- **Status checks**
  - Periodic or event driven
  - For POLICY_DATA arriving in any message type
- **Error processing**

Action Items

- **Agree on RSVPs extensions:**
  - Policy objects
  - Default handling
  - Semantic fragmentation
  - Resv Report message (also ack?).
  - RSVP/policy interface principles
- **Immediate Future Work**
  - RSVP/policy interface: detailed design
  - LPM mechanisms: processing rules
  - Hear about policies / other models from the WG
    - Where (document) should policy attributes be defined