Common Endpoint Locator Pools (CELP)

- **draft-crocker-celp**
  - Dave Crocker
  - Avri Doria

- **Multiple multiaddressing schemes**
  - Different approaches have different benefits

- **Proposal:**
  - Share locator pools across independent associations
  - Reduce multiaddressing control transaction costs
  - Improve availability of locator performance information
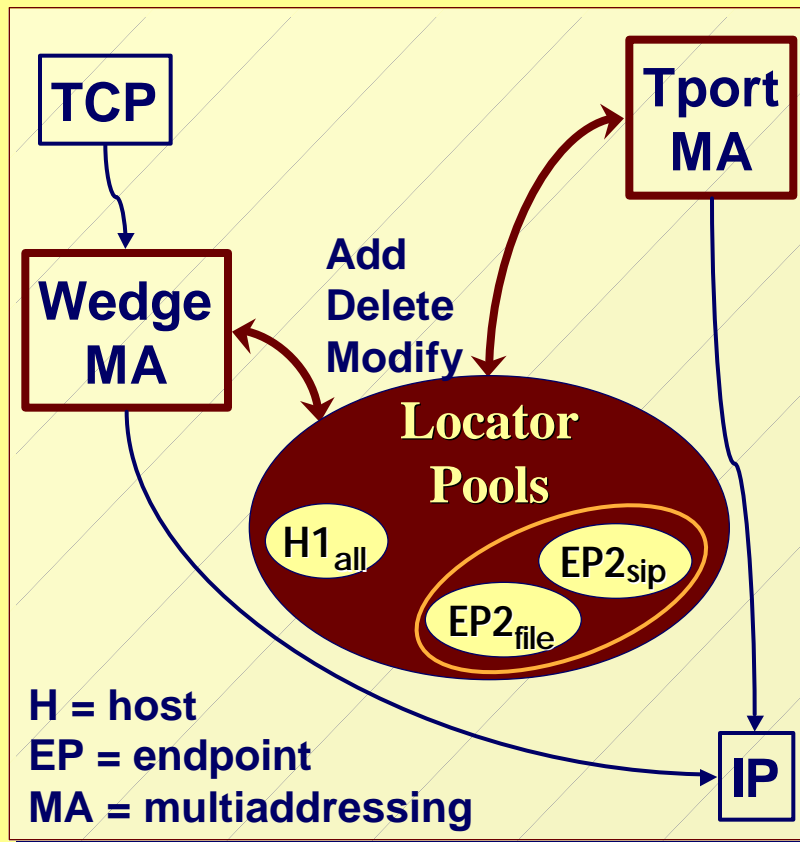
Synergy Across Associations

- **Transport-based schemes**
  - Multiplex control exchange in data stream, so control data does not increase packet overhead
  - Can naturally obtain path quality information

- **Wedge-based schemes**
  - Multiaddressing for legacy transports
  - Naturally independent of individual transport associations
  - Can operate asynchronously of associations, deferring control exchanges, often needing no exchange
  - Can maintain pools with different referential granularity

Framework

TCP

Tport MA

Wedge MA

Add
Delete
Modify

Locator Pools

H1$_{all}$

EP2$_{sip}$

EP2$_{file}$

IP

H = host
EP = endpoint
MA = multiaddressing

° **Variable granularity**
  - {local, remote}
  - {local, remote, flow}
  - {local, remote, protocol, port}
  - {local, remote, type of service}

° **Status**
  - Reachability
  - Performance

Issues

- **Path selection**
  - Which paths are available or better?
  - Suggestion: *Defer generality; start with primary and fallback choices*

- **Local/Remote combinatorials**
  - Suggestion: *Defer generality; start with just {remote} or {local, remote}*

- **Security**
  - Different schemes have different degrees of security → concern about weakest participant affects entire service
  - Maintaining synchrony among different modifiers of pool

- **Referential commonality**
  - Different schemes use different identifiers
  - How to know that different locators refer to same endpoint?
  - Suggestion: *That's what domain names are for…*

Next Steps

- **Resolve**
  - Differential security issues
  - Near-term vs. long-term issues
  - Determining common endpoint referencing

- **Formulate CELP service model details**
  - Data structures
  - Operations