LEDBAT architecture framework consisting of pluggable components

draft-mayutan-ledbat-congestionarchitecture-00.txt

Mayutan Arumaithurai, Xiaoming Fu, K.K Ramakrishnan

March 22, 2010
LEDBAT design goals

1. Saturate bottleneck
2. Keep delay low
3. Yield to traffic using standard TCP
4. Add little to queuing delays
5. Operate well with FIFO and DROP tail queues
6. Be deployable for popular applications
7. Use ECN, AQM, DiffServ where applicable
Figure: Architecture consisting of pluggable components
LEDBAT Protocol Suite

Figure: Architecture consisting of pluggable components
**Figure:** Architecture consisting of pluggable components
Figure: Architecture consisting of pluggable components

Each module operates in a different timescale
Congestion Detection Module

- **Delay Based**
  - + Does not require network support
  - − Sensitive to variation in routes, bottleneck buffer size, bursty traffic etc.
- **Loss based**
  - + Reliable indicator of congestion
  - − Results in substantial interference to TCP
- **ECN marking based**
  - + Good and early indicator of the onset of congestion
  - − Requires network support
- **Delay + Loss/marking based**
Congestion Detection Module

- Delay Based
  - + Does not require network support
  - − Sensitive to variation in routes, bottleneck buffer size, bursty traffic etc.
- Loss based
  - + Reliable indicator of congestion
  - − Results in substantial interference to TCP
- ECN marking based
  - + Good and early indicator of the onset of congestion
  - − Requires network support
- Delay + Loss/marking based

Congestion indicator:
- Binary states: congested or non-congested
- Multiple levels: 0, 0.1, .., 0.5, .., 1
Flow Control Module

- Standard TCP (AIMD)
  - + Robust: Good indication of available capacity
  - – Substantial queuing, thereby delay
  - – Conservative in using available bandwidth

- Variants (Aggressive Increase)
  - + Good for high BDP networks
    - Without bandwidth estimation
      - – Cause interference: No prior knowledge of available bandwidth
    - With Bandwidth Estimation
      - + Separates congestion control from bandwidth estimation
      - – Slower
      - – Involves additional overhead

=> Always necessary to have an estimate of available bandwidth
=> More useful in the context of LEDBAT, due to submissive nature
Flow Control Module

- **Standard TCP (AIMD)**
  - + Robust: Good indication of available capacity
  - − Substantial queuing, thereby delay
  - − Conservative in using available bandwidth

- **Variants (Aggressive Increase)**
  - + Good for high BDP networks
    - Without bandwidth estimation
      - − Cause interference: No prior knowledge of available bandwidth
    - With Bandwidth Estimation
      - + Separates congestion control from bandwidth estimation
      - − Slower
      - − Involves additional overhead

=> Always necessary to have an estimate of available bandwidth
Flow Control Module

- Standard TCP (AIMD)
  - + Robust: Good indication of available capacity
  - - Substantial queuing, thereby delay
  - - Conservative in using available bandwidth

- Variants (Aggressive Increase)
  - + Good for high BDP networks
    - Without bandwidth estimation
      - - Cause interference: No prior knowledge of available bandwidth
    - With Bandwidth Estimation
      - + Separates congestion control from bandwidth estimation
      - - Slower
      - - Involves additional overhead

=> Always necessary to have an estimate of available bandwidth

=> More useful in the context of LEDBAT, due to submissive nature
Bandwidth Estimation Module

- Standard TCP (increase until loss)
- Delay based (e.g. Vegas, Compound TCP)
- Probing based
- Router assisted (e.g. Quick start)
- Support of some oracle server
Figure: LEDBAT example with varying Congestion detection components
Figure: LEDBAT example with varying flow control components
Figure: LEDBAT example with varying bandwidth estimation components
We could use it as a guideline while standardizing a CC mechanism to keep it flexible.

- Each module and component can be independently standardized
  - Decoupling each module
- Often implicitly followed in current specifications