Paper survey related with web/app performance optimization and MEC

Youngseok Lee

lee@cnu.ac.kr

cnu.lee@ucdavis.edu

1. Mobile Edge Computing: A Survey, in *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 450-465, Feb. 2018.
2. Optimization of Webpage Downloading Performance with Content-aware Mobile Edge Computing. In Proceedings of the Workshop on Mobile Edge Communications (MECOMM '17). ACM
3. Enabling context-aware HTTP with mobile edge hint, *2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, Las Vegas, NV, 2017,

Revisiting MEC

- MEC architecture is a new revenue stream for mobile operators that has not matured sufficiently
- A few application areas adopting edge computing
  - Fog computing, AR, content delivery

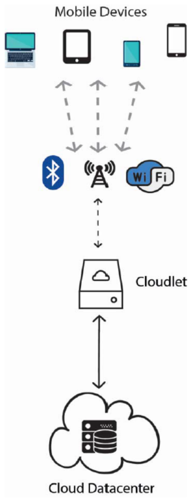


Fig. 1. Cloudlet.

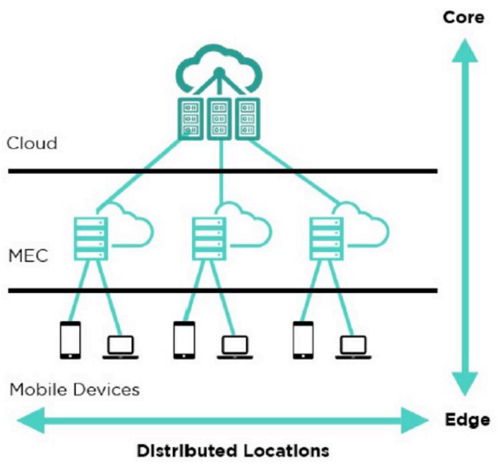


Fig. 3. Three-layer architecture.

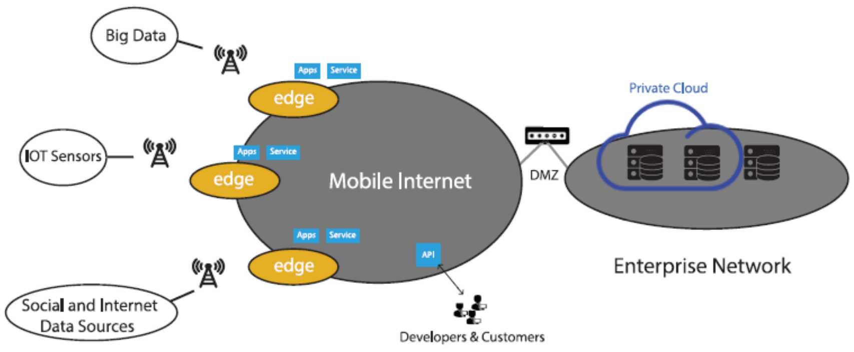


Fig. 4. MEC architecture.

Research areas

- Computation offloading
  - Edge accelerated Web browsing (EAB) prototype designed for Web application execution using a better offloading technique.
  - Network architecture-based solutions, such as cloud of co-located mobile devices
- Storage
- Low latency
- Energy efficiency

Research on Infrastructure

- Deployment scenarios
  - MECs in outdoor: RAN
  - MECs in indoor: Wifi or 3G/4G access points
- MEC testbeds
  - 5G test network at Oulu, Finland
  - Industrial testbeds: Nokia and China

Other Open Issues

- Security
  - The application data movement: possible with encryption
- Pricing

- Web interface
  - Not optimized for mobile
- Other
  - Privacy, openness, multiservices, robustness, resilience

Content Delivery and Caching

- The edge computing technology plays a key role in website performance optimization
  - caching HTML content
  - reorganizing Web layout
  - resizing Web components

Improving Short-lived Web Traffic Performance

- How to compensate the throughput gap caused by the computation latency during short-lived application loading by only adapting the transport-layer protocol?
  - not affected by any application layer constraints such as HTTPs content encryption and security policy
  - embed network intelligence at mobile edge
- Through the optimization of TCP initial window (IW) size
  - short-lived applications such as webpage downloading, where content downloading is normally completed during the TCP slow-start phase
- Mobile edge's awareness of the computation time on the device
  - does play an important role in webpage downloading performance

MEC Support

- Optimize TCP IW
- DNS response with context information

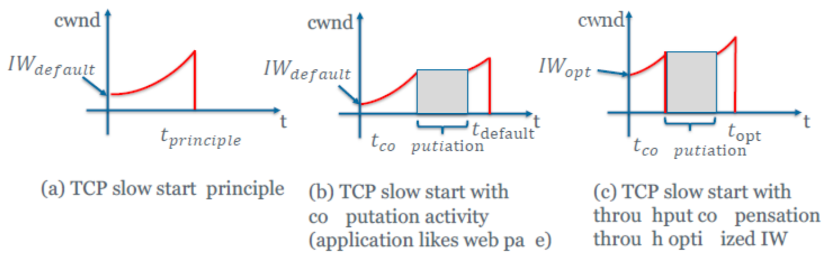


Figure 2: Throughput gap due to computation activity

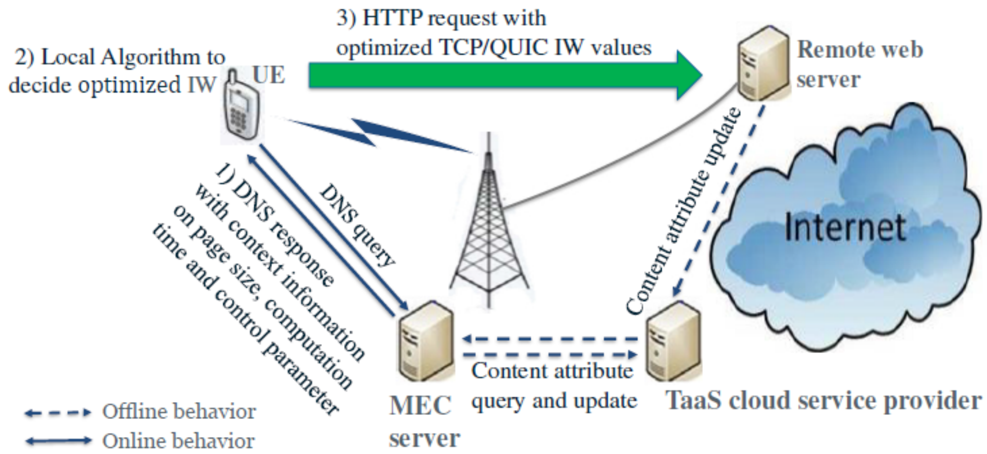


Figure 1: Overview of proposed MEC framework

How MEC Server Obtains Context Info

- Not every web content but popular web pages
- Testing as a service (TaaS)
  - MEC server performs test/measurement services
  - Google firebase or Flywheel
  - Computation latency and total size of content can be obtained

Performance Evaluation

- real LTE-A testing infrastructure
- use QUIC as the underlying protocol
- when the computation latency accounts for less than 20% of the overall downloading time,
  - the throughput gap can be fully compensated.

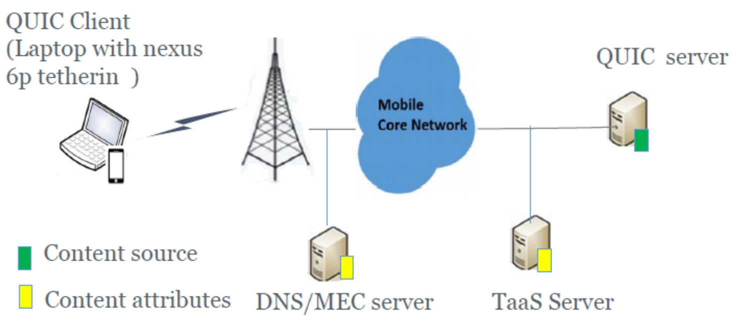
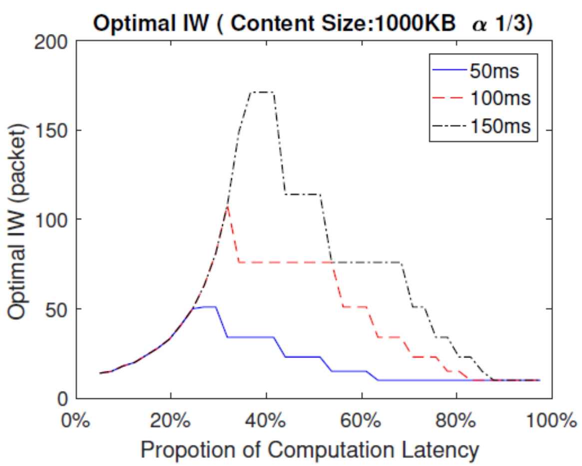


Figure 3: Proof-of-concept implementation in LTE-A test bed

- When the proportion of computation latency varies between 20% and 50%,
  - the throughput can be improved up to 34.5%. Such improved downloading throughput has led to the reduced webpage downloading time by up to 25.1%.



5: Optimal IW in real LTE-A network (varying content size)

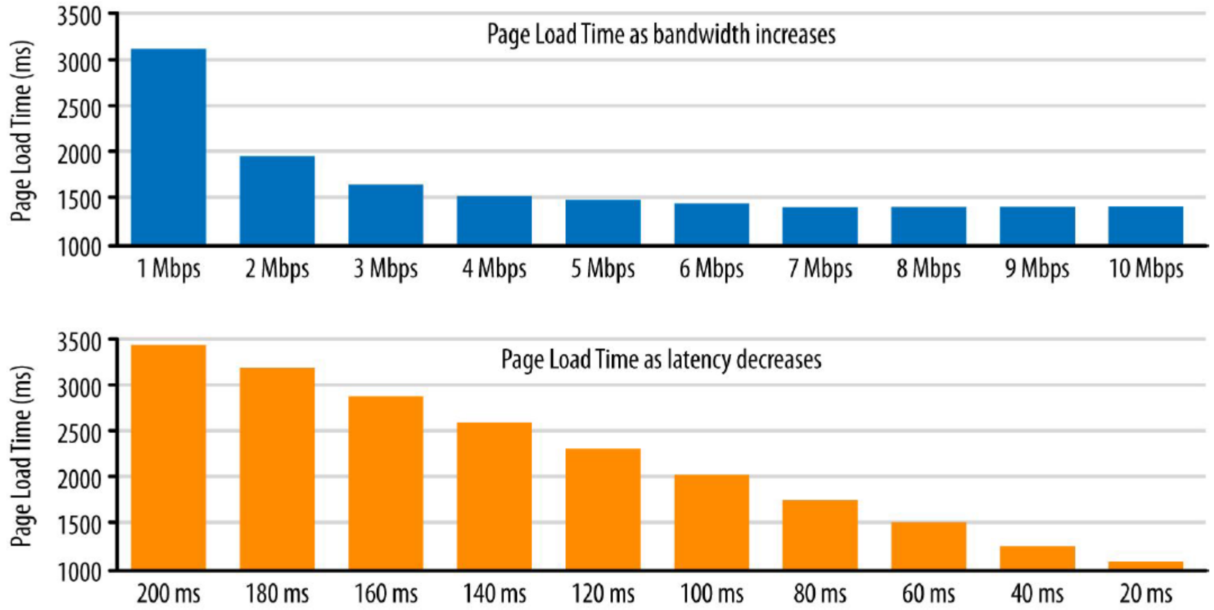
Recent Changes in Web

- HTTPS
  - TLS + HTTP
- QUIC
  - UDP + TLS + TCP Congestion Window algorithm
- HTTP/2: low latency protocol
  - Single TCP connection, Server push, Header compression, binary, multiple streams

```
yslee@yslee-ThinkPad-T410 ~ $ sudo ss -ti
         cubic wscale:9,7 rto:392 rtt:145.399/40.693 ato:40 mss:1428 cwnd:10 bytes_acked:1688 bytes_received:382
1 segs_out:10 segs_in:10 send 785.7Kbps lastsnd:655240 lastrcv:595040 lastack:595040 pacing_rate 1.6Mbps rcv_rtt
:144 rcv_space:29200
```

- TCP initial window
  - IETF RFC 6928 – a proposal to increase the TCP Initial Window to 10 segments: 10 * 1432 bytes = 14KB for the initial web page

Latency vs Bandwidth impact on Page Load Time



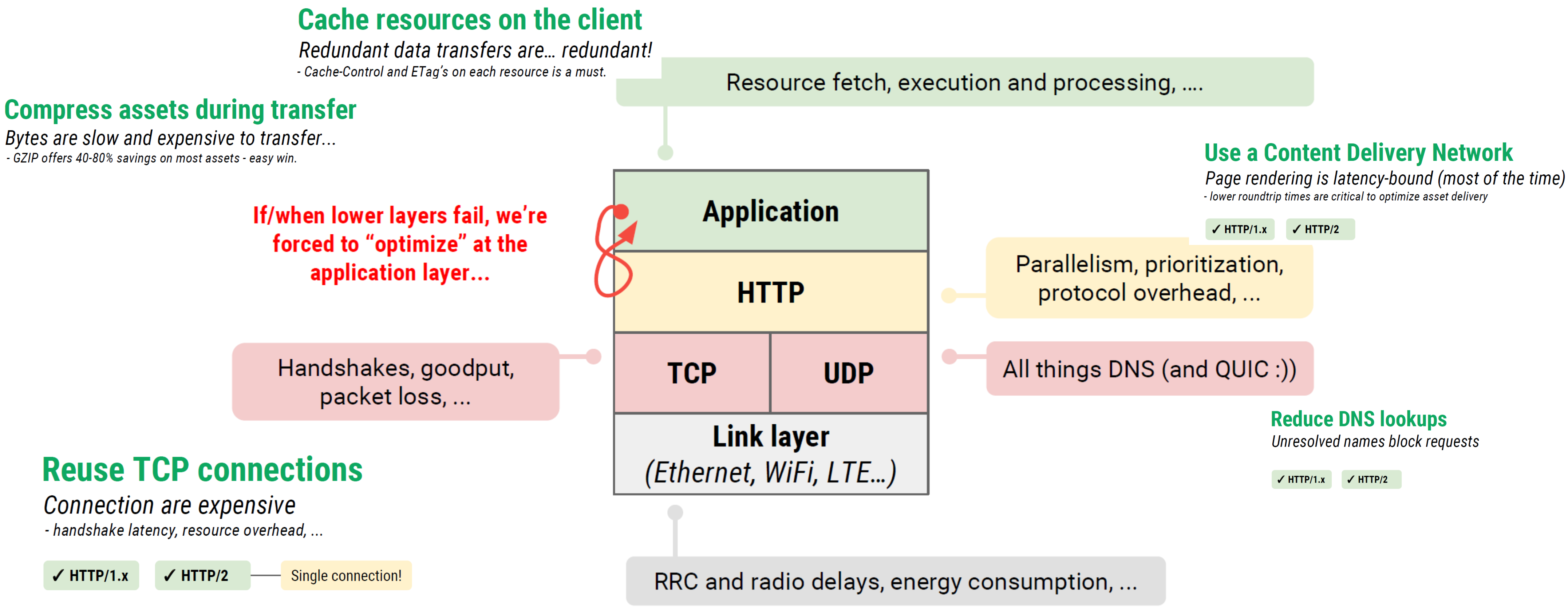
Single digit % perf improvement after 5 Mbps

Linear improvement in page load time!

"**To speed up the Internet at large, we should look for more ways to bring down RTT.** What if we could reduce cross-atlantic RTTs from 150 ms to 100 ms? This would have a larger effect on the speed of the internet than increasing a user's bandwidth from 3.9 Mbps to 10 Mbps or even 1 Gbps." - Mike Belshe

bit.ly/http2-opt

Optimizing Web



bit.ly/http2-opt

Questions about supporting mobile web/app performance by MEC server

- TCP configuration?
  - Scalability issue for every web domain and TCP connections
  - Security issue for kernel-level TCP configuration
    - Root privilege is required
- QUIC configuration
  - Application layer configuration: CUBIC + TLS + UDP
  - Only by Google servers

Discussion

- Improving end-to-end application performance by middle box (MEC/proxy/CDN)
  - Challenges
    - Encryption: TLS, (DNS)
    - Security: certificate pinning, HSTS
    - Scalability
  - Possibilities
    - Caching/proxy: Amazon Silk, Opera mini
    - CDN: Akamai/Limelight, Netflix
    - HTTP/2 optimization with TaaS
      - DNS, server push, compression, concatenating resources, inline resources

Speeding up Web Page Loads with Shandian, USENIX 2016