Experimental Evaluation of Transport Services CoAP, HTTP and SPDY for Internet of Things

Laila Daniel, Markku Kojo and Mikael Latvala*

Department of Computer Science
University of Helsinki
* Mosa Consulting, Finland

September 23, 2014

Overview of the presentation

Transport Services for IoT

- To extend Internet services to the IoT devices a suitable transport service is needed
- The transport service should be
  - Compatible with the TCP/IP suite
  - Open standard and proven to be scalable
  - Efficient in the resource constrained IoT environment

- In this study we consider three transport services
  - HTTP (HyperText Transfer Protocol)
  - SPDY (pronounced as Speedy)
  - CoAP (Constrained Application Protocol)
- How close to CoAP can HTTP/SPDY be ?
- Do IoT devices that are less constrained require very light-weight transport service like CoAP ?

HTTP and SPDY

HTTP

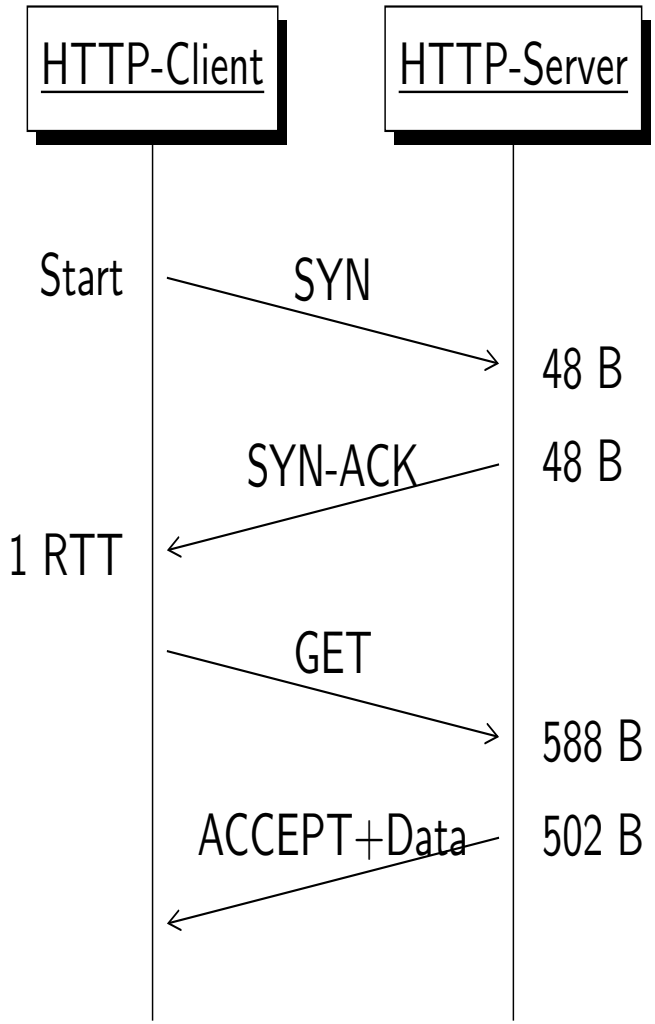
- De-facto standard of information transfer in the Internet
- TCP is the transport protocol for HTTP
- Problems of HTTP in IoT environment
  - Lengthy HTTP headers, For smaller packets, the protocol header overhead is around 150 %
  - The need to establish TCP sessions for each request-reply data transfer. For short data transfers, TCP connection establishment time will be up to 50% of the total transfer time

SPDY

- Originally from Google to make HTTP faster, basis for HTTP/2
- Allows many concurrent HTTP requests in a single SPDY session over a single TCP connection
- Similar problems as that of HTTP

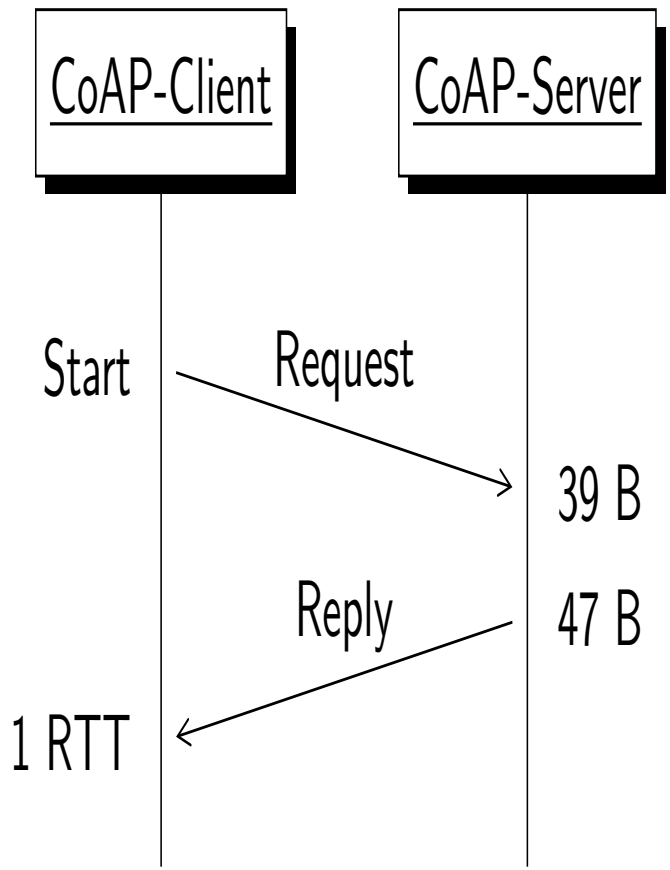
HTTP: Downloading a 10-byte object - phases of data transfer

- Legacy browser is used
- TCP connection establishment
- GET request
  - Get ./10 HTTP/1.1
  - Details of the browser, OS
  - Accepted data types, encoding and encryption schemes
- ACCEPT + Data
  - HTTP 1.1/ 200 OK
  - Date and time
  - Server details, name, last updated
- Total bytes transferred 1546 bytes including TCP ACKs



CoAP: Downloading a 10-byte object phases of data transfer

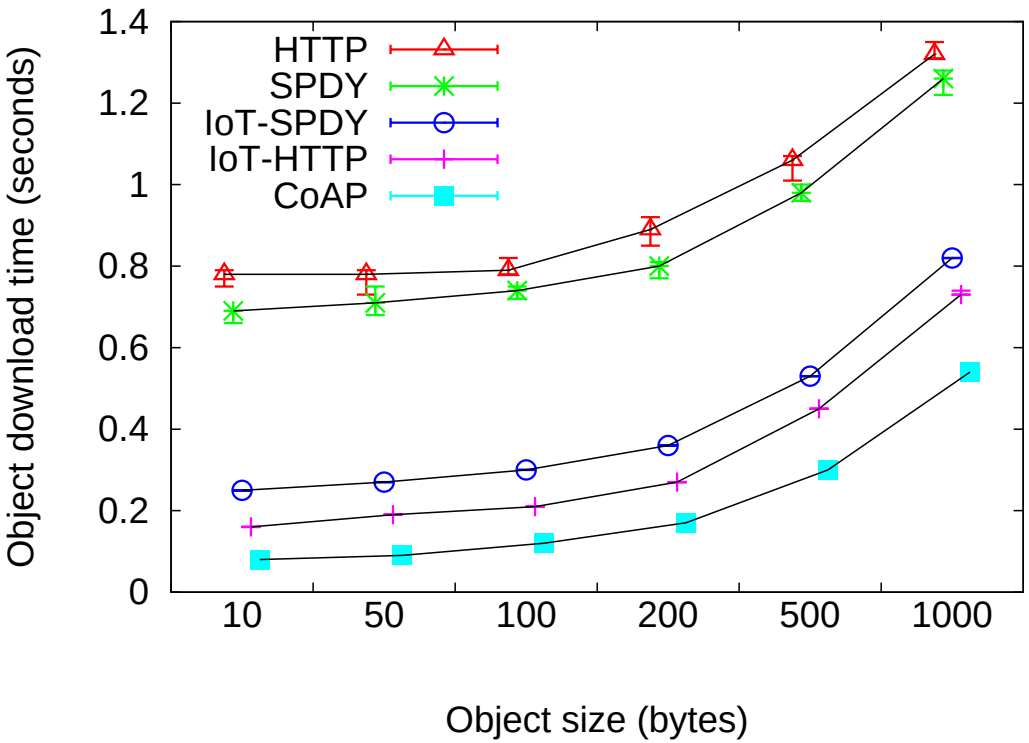
- Specifically designed for constrained devices and constrained networks
- HTTP-CoAP proxy to connect to the Internet, scalability issues
- Operates in request-response mode
- Short binary header of 4-bytes
- UDP is the transport protocol, so no connection establishment phase
- Supports optional reliability
- Problems in IoT environment
  - Error-prone and/ or congested paths, slow recovery of packets



How to make HTTP and SPDY better suited for IoT?

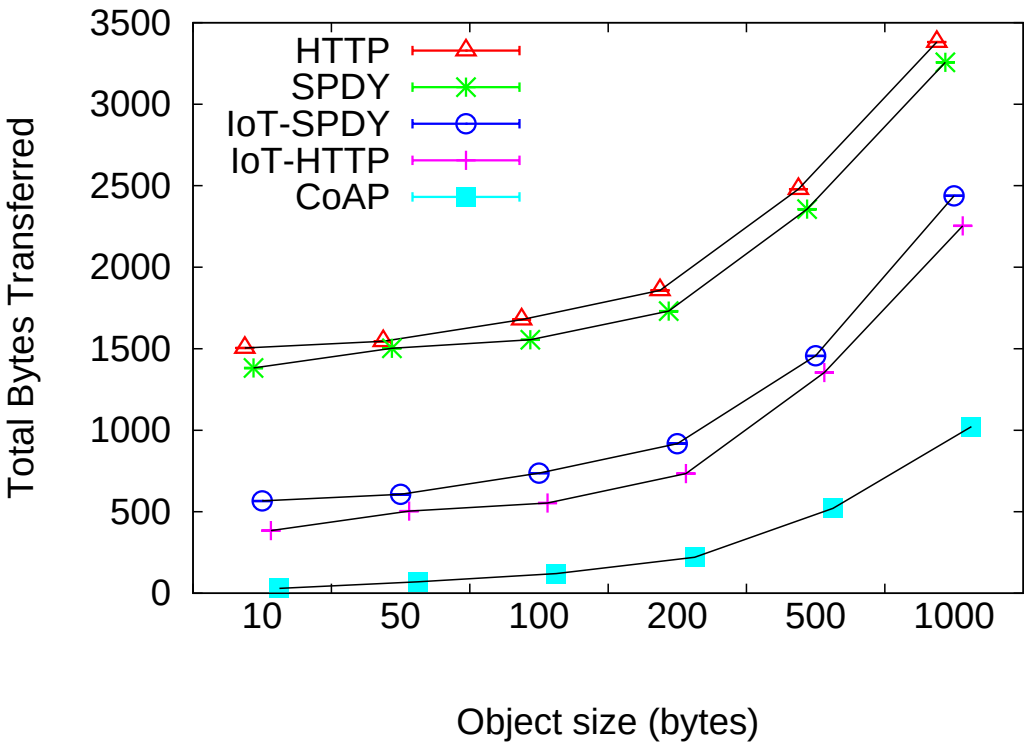
- Minimize the connection establishment time
- Reduce the size of the headers associated with HTTP and SPDY, many headers are not really needed
- We introduce IoT-HTTP and IoT-SPDY
  - Adaptations of HTTP and SPDY for IoT environment
  - Able to send request at the TCP connection establishment phase using TCP Fast Open (TFO)
  - Reduced headers

Object Download time



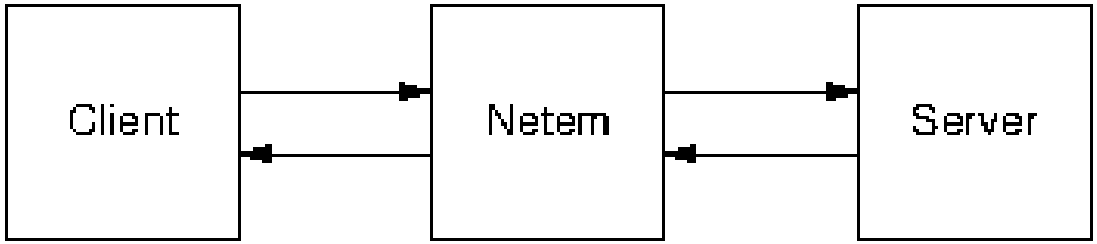
- Object download time : time duration between the client request and the arrival of the final byte of the object data

Total bytes transferred to fetch an object



- Total bytes transferred includes all headers

Experimental Setup



- Linux Netem emulates IoT link of data rate 20 Kbps and link delay of 20 ms. Link MTU 128 bytes.
- The link characteristics roughly corresponds to Zigbee
- Error-free links

Transport Services used in the experiments

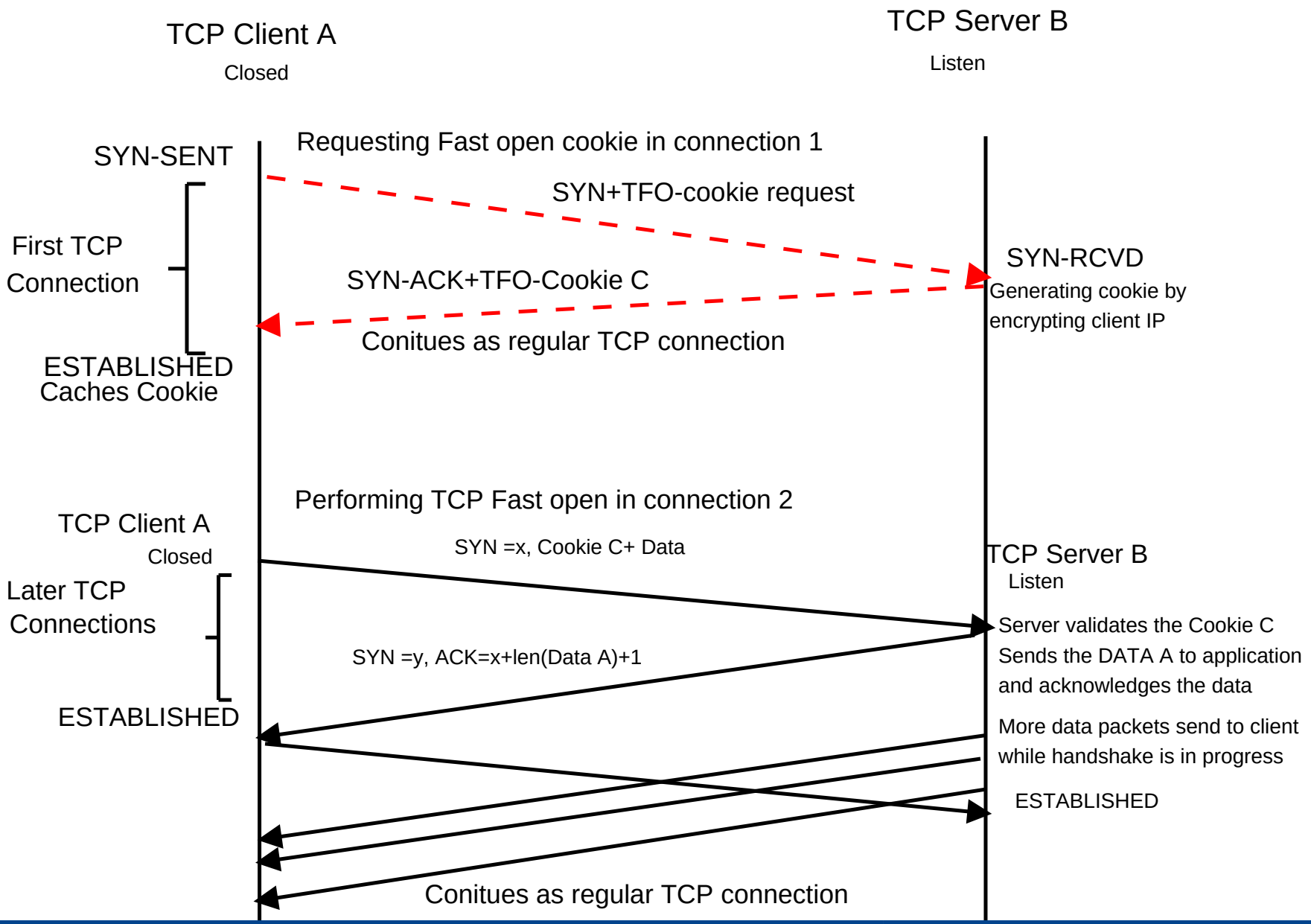
- CoAP - libcoap client and server (libcoap)
- HTTP - Google Chrome client and Apache server
- SPDY - Google Chrome client and Apache server with mod-spdy enabled
- IoT-HTTP - A simple Web server and client + TFO
- IoT-SPDY - spdyd server and spdy-python client+TFO
  Object sizes - 10B, 50B, 100B, 200B, 500B and 1000B

TCP Fast Open (TFO)

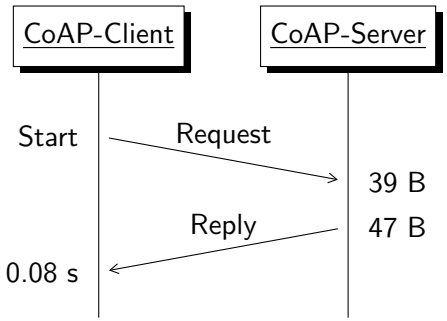
- TFO enables safe data exchange during TCPs initial handshake
- Decreases application network latency by full RTT, decreasing the delay experienced by short TCP transfers
- Server uses a security cookie to authenticate a client initiating a TFO connection, so no need of 3-way handshake for additional TCP connections between the same hosts

- Internet Draft - draft-cheng-tcpm-fastopen-09.txt. Y. Cheng, J. Chu, S. Radhakrishnan and A. Jain.
- A paper from the authors in ACM CoNEXT 2011

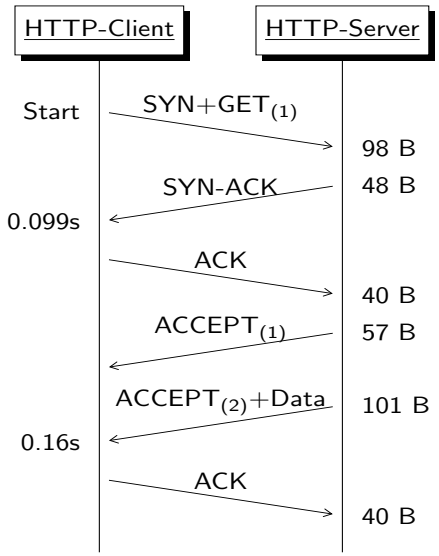
TCP Fast Open (TFO)



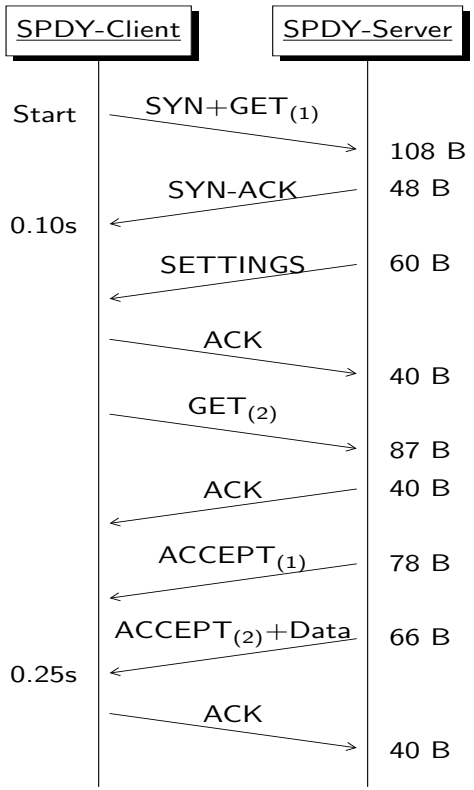
Message Sequence Chart for 10-bytes object



CoAP



IoT-HTTP



IoT-SPDY

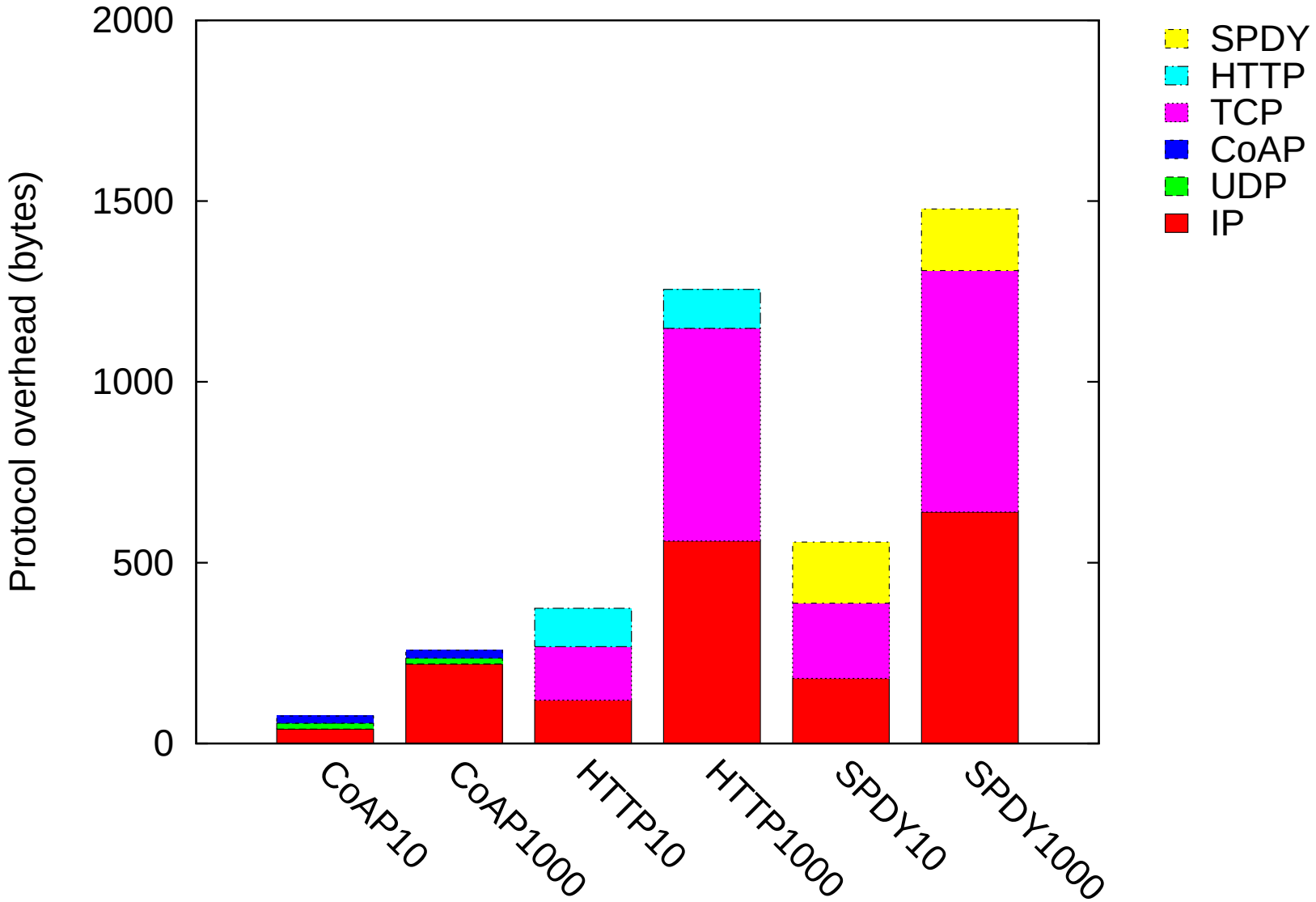
Summary of analysis of downloading a 10-byte object

Network environment

- Zigbee-like environment
- Link data rate 20 Kbps, link MTU 128 bytes
- Error-free links

Metrics	CoAP	IoT-HTTP	IoT-SPDY	SPDY	HTTP
Object download time	0.08s	0.16s	0.25s	0.68s	0.79s
#Packets	2	6	9	20	20
TotalBytes	86	384	567	1382	1546

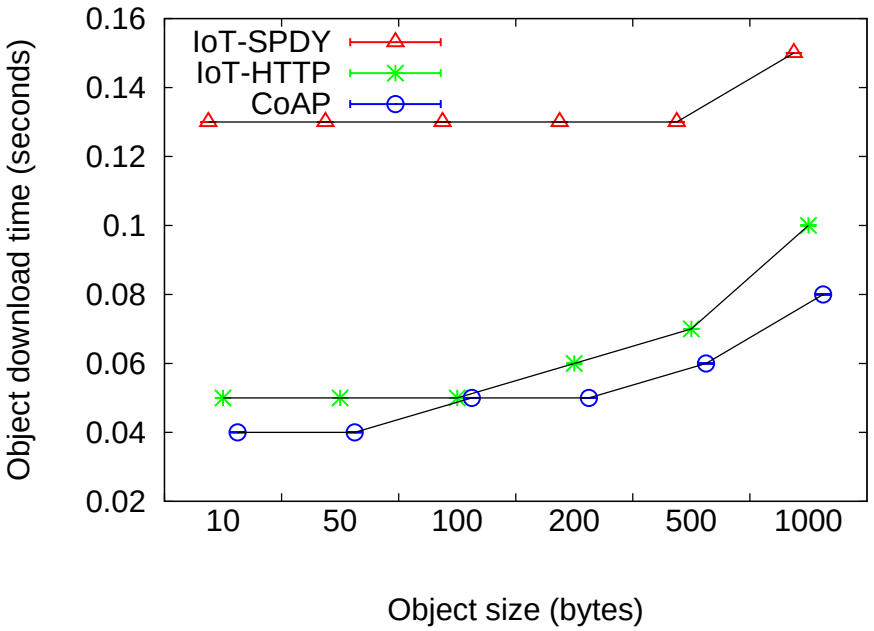
Protocol Overhead: CoAP, IoT-HTTP and IoT-SPDY



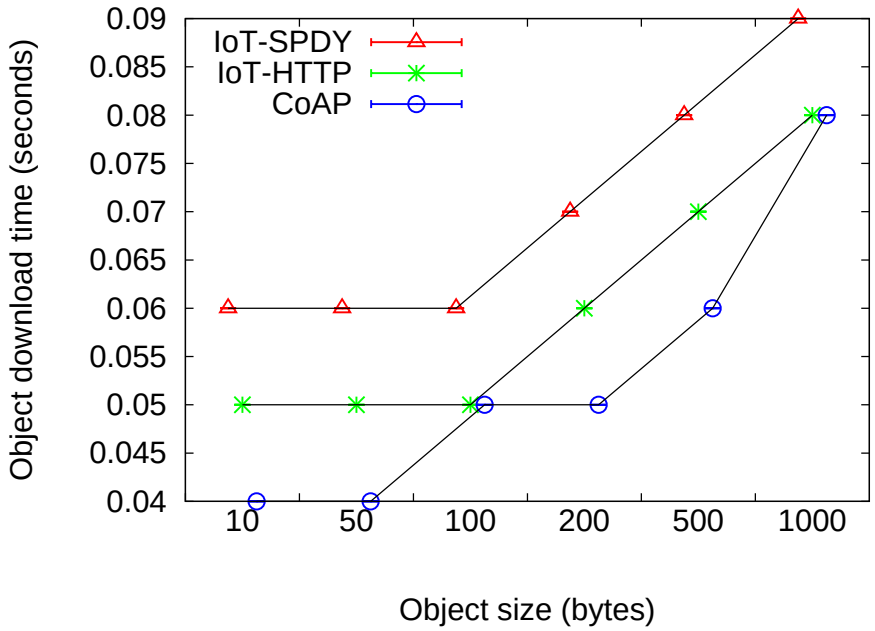
Protocol Overhead: CoAP, IoT-HTTP and IoT-SPDY

	Transport Services	Protocol overhead					
		IP	UDP	TCP	CoAP	HTTP	SPDY
10-byte object	CoAP	40	16		20		
	HTTP	120		148		106	
	SPDY	180		208			169
1000-byte object	CoAP	220	16		22		
	HTTP	560		588		108	
	SPDY	640		668			170

Object Download time

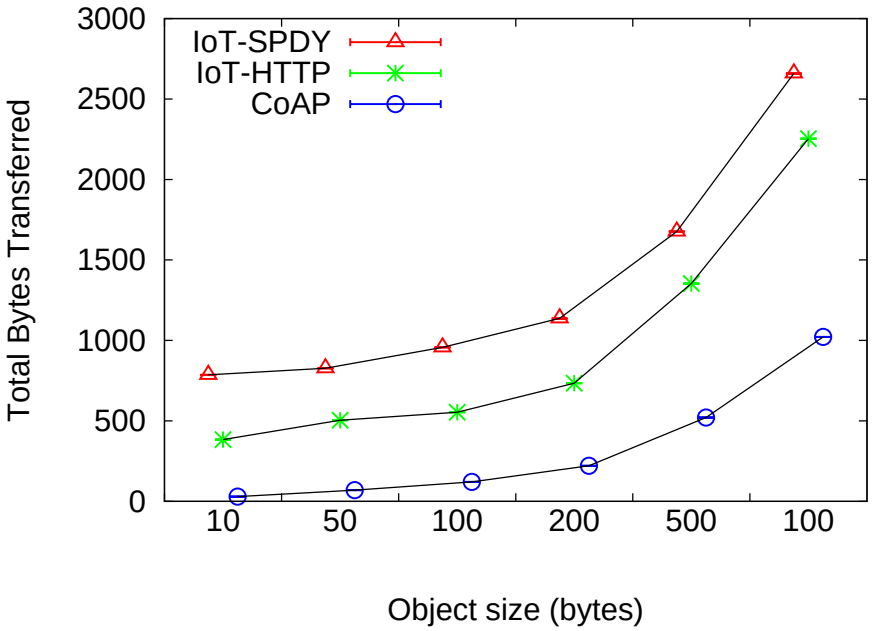


Data rate 250 Kbps, MTU 128 B

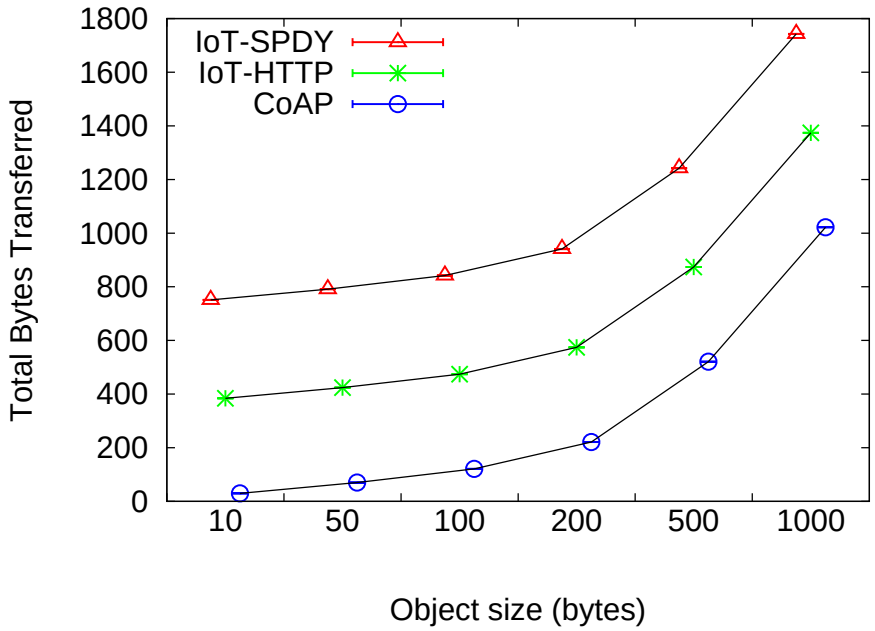


Data rate 250 Kbps, MTU 1280 B

Total bytes transferred to fetch an object



Data rate 20 Kbps, MTU 128 B



Data rate 250 Kbps, MTU 1280 B

Conclusions

- CoAP is the best in constrained environments with no link errors
- IoT-HTTP and IoT-SPDY reduce the object download time by 50 % to 75 % compared to HTTP and SPDY
- When data rate increases, object download times for CoAP and IoT-HTTP are comparable
- When MTU size increases, total bytes transferred by HTTP and SPDY decrease
- In not so constrained environments, IoT-HTTP and IoT-SPDY can be used

Plans for future experiments

- Study the behaviour of CoAP, IoT-HTTP and IoT-SPDY in error-prone/congested IoT environments
- Improve SPDYs compression techniques
- Incorporate RObust Header Compression (ROHC) with TCP
- Further TCP enhancements

- Thank You for your attention!
- Any questions, comments ?