Exam Hints

Exam Hints (1)

▶ Friday, 15 Dec 2006, 13 – 16, S5

▶ There is a ton of paper you **could** read
  - Particularly RFCs, Internet drafts, etc.
  - But this would simply be too much.

▶ What you **SHOULD** do includes
  - Understand all the slides from the lectures
  - Read the overview parts of RTP, SAP, SDP, RTSP, MSRP, MRCP, SIP, ICE
    - Need a good grasp of the big picture of the respective protocols
  - If there questions about some core aspects, look them up
    - E.g., if the semantics of the Expires: header in the REGISTER message is unclear
    - E.g., if you don't know the purpose of a SIP Request URI
    - E.g., if you wonder what an RTSP session is and how it is created and destroyed
  - There are too many details: concentrate on those discussed in the lecture
    - E.g., there are many error codes and additional headers in SIP we did not talk about

Exam Hints (2)

▶ Primary sources: RFCs and Internet Drafts
- http://www.ietf.org/rfc.html
- http://www.ietf.org/iesg/1rfc_index.txt

- http://www.ietf.org/ID.html
- https://datatracker.ietf.org/public/idindex.cgi

▶ Working groups
- http://www.ietf.org/html.charters/avt-charter.html
- http://www.ietf.org/html.charters/sip-charter.html
- http://www.ietf.org/html.charters/sipping-charter.html
- http://www.ietf.org/html.charters/simple-charter.html
- http://www.ietf.org/html.charters/mmusic-charter.html
- http://www.ietf.org/html.charters/speechsc-charter.html
- http://www.ietf.org/html.charters/iptel-charter.html
- http://www.ietf.org/html.charters/geopriv-charter.html
- http://www.ietf.org/html.charters/xcon-charter.html

Exam Hints (3)

▶ Planning on 10 – 12 questions

▶ Questions will be about **concepts rather than details**
- Concepts obviously include

  Architecture, general operation and interactions, terminology, methods, and headers, basics of message exchanges and packet flows
- Concepts do not include

  Syntax details, tiny exceptions, state machines, long call flows in lots of detail, numbers of response codes…

▶ May include a **small design task**
- How would you build a system that does X?
- Where to get which data from?
- Which protocols to apply? How to combine them?
- May leverage what you have learned in the assignments

Exam Hints (3)

- ▶ Range: All lectures except for the "Real World SIP" part today
- ▶ Things learned when looking closer at the exercises
- ▶ Again: concepts rather than details
  - But going once through all the slides will likely be insufficient
  - So, take your time

- ▶ Task structure
  - 10 – 12 in total
  - Large fraction with (relatively) short answers (6 points each)
  - 2 – 4 requiring more time (6 points each)
  - Probably one "design" task (6 or 12 points)

- ▶ Some sample questions (*probably* not used in the exam :-)

Sample Questions

Short tasks (type A)

- ▶ How is a SIP transaction identified?

- ▶ Why do RTP packets carry a sequence number *and* a timestamp?

- ▶ Why is jitter not a problem for real-time communications in packet networks? What *is* the problem?

- ▶ What are the IMG FETCH and RESOLVE operations used for?

- ▶ What is the media level a=rtpmap attribute in SDP used for?

- ▶ Sketch the operation of SIP digest authentication.

Sample Questions (2)

Longer tasks (type B)

▸ Sketch the interaction of RTP and RTCP for synchronizing two media streams (e.g., audio and video) from the same source.

▸ Outline the operation of the SIP REGISTER messages. Which different semantics are supported? Which parameters are used to control these semantics?

▸ What is the basic idea of audio redundancy encoding? Contrast this approach to generic FEC, e.g., for use with video.

▸ What are the semantics of the following RTSP message? Describe the key fields. When will it be sent? Who will send it?

Sample Questions (3)

Design tasks (type C)

(likely to revolve somehow around service creation)

▸ Sketch one approach (out of many possible ones) to realize a call recording feature for a SIP user who uses a SIP hardware phone without built-in recording capabilities. Remember that this service must not require cooperation from the remote party on a call. Describe which components you will use, which functions they perform and when and how they interact (protocols, messages).

Any other Questions…?