Summing up: How the Internet Works

- Important protocols we haven't got time for
  - We haven't said nearly enough about security
- How things fit together
- Guiding principles
- Questions?

Other infrastructure topics

- PPP (point-to-point protocol)

- EAP, RADIUS, DIAMETER
  - Authentication, authorisation

- IPSec, IKE (Shay 11.3)
  - Applies to IPv4 *or* IPv6

- VPN (virtual private networks)

- NAT
  - Network address translation

- Firewalls

- SOCKS (firewall traversal)

- Multicast (Shay 11.2)

- Mobile IP, mobility in general

- SASL (simple auth & security)

- SLP (service location)

- RSVP (Shay 11.2)

- ROHC (header compression)

- iSCSI (SCSI over IP)

- RDMA (remote DMA)

Other application topics

- MIME (multimedia formats)
- SIP, ENUM
  - standards for voice over IP
- Video over IP
- PGP, S/MIME (secure email)
- Internationalised email
- Anti-spam solutions
- LDAP (directory)
- NTP (network time protocol)
- IPP (Internet printing protocol)

- NFS, AFS
  - Remote file systems
- NNTP (network news)
- RSS, ATOMPUB (feeds)
- Instant messaging
- Language tags
- Web Services
  - XML-based distributed computing over SOAP+HTTP
- Peer to Peer protocols
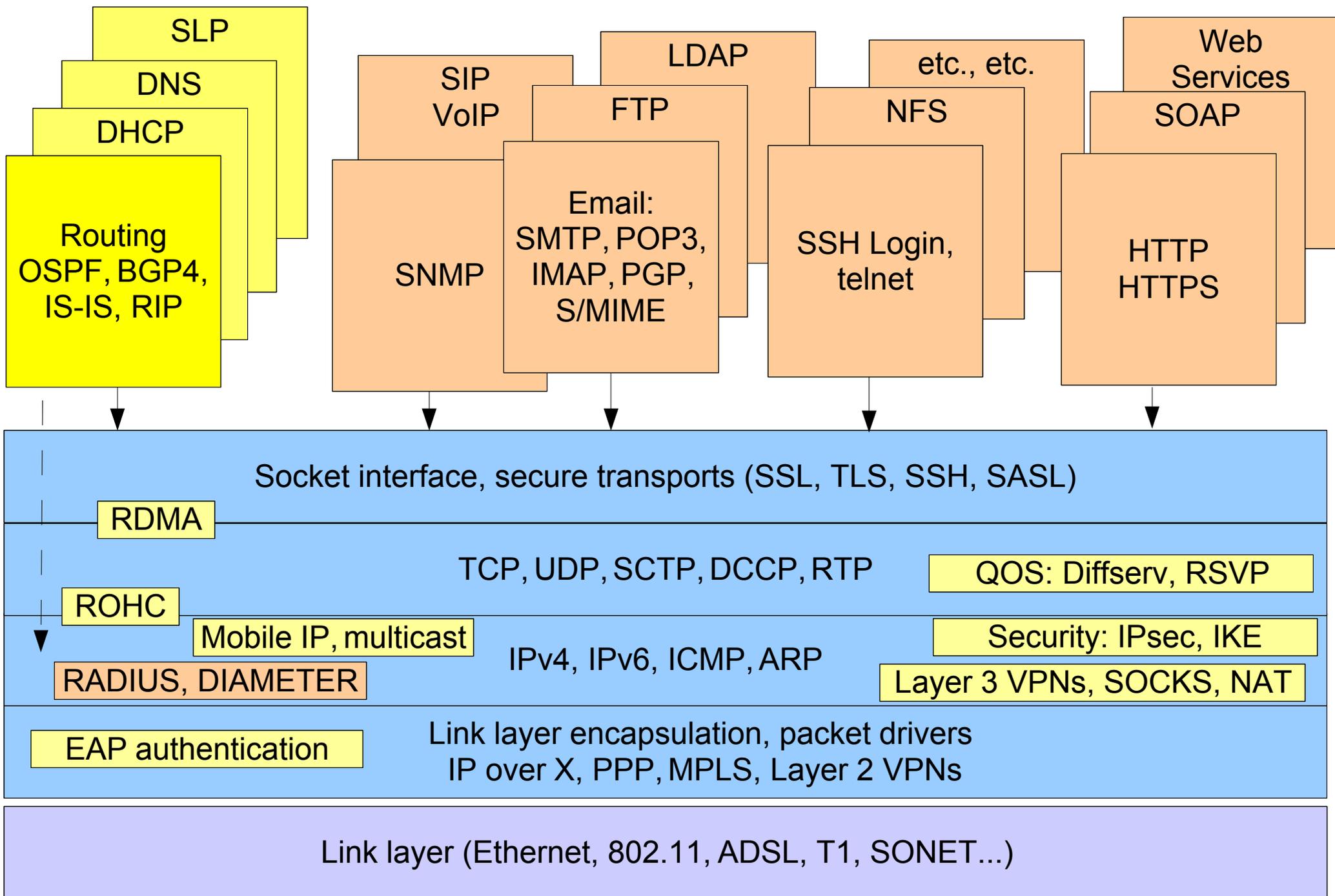- Grid computing protocols

The kitchen sink - a list of topics

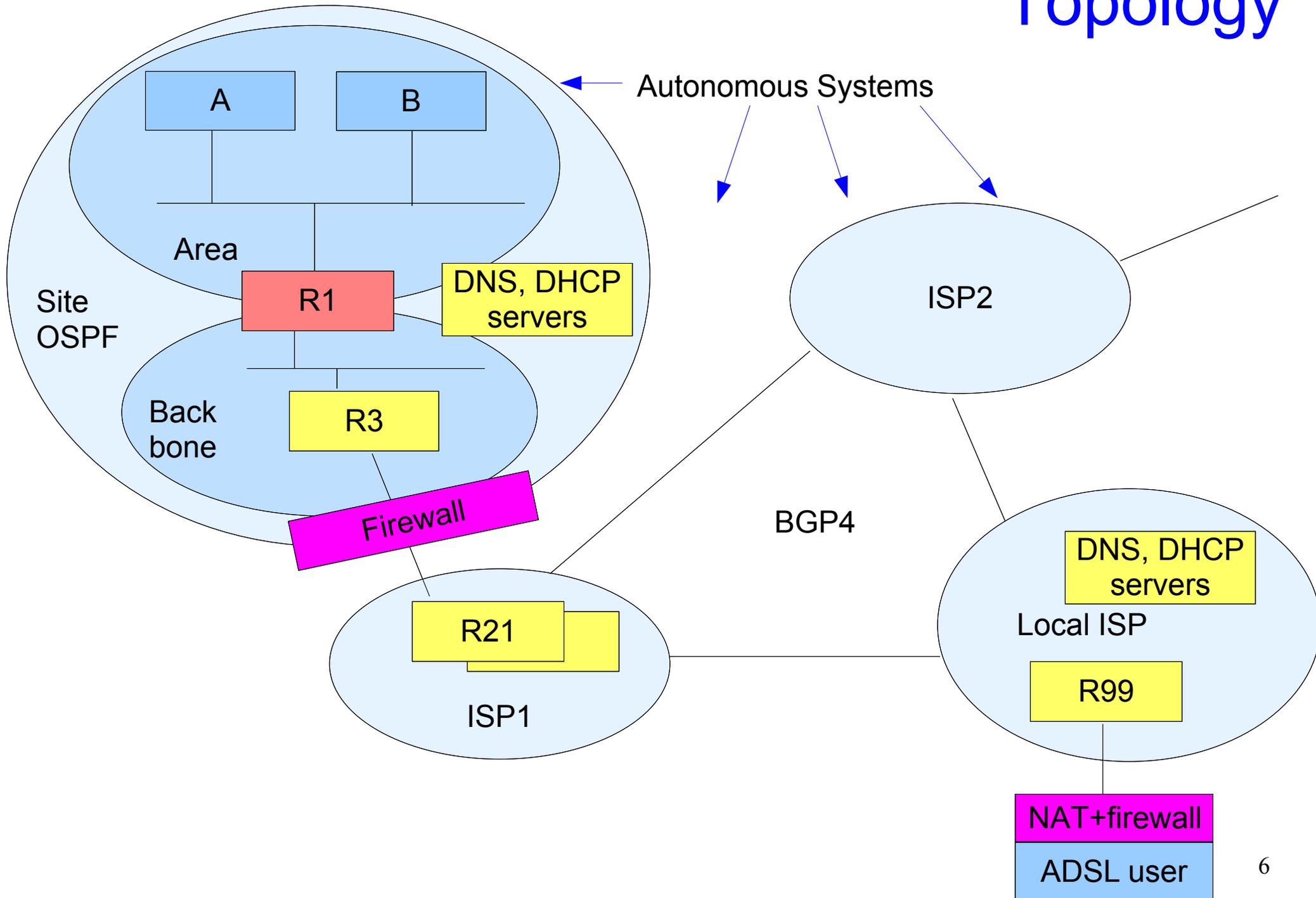- This is only to illustrate the complexity and richness of Internet protocols; don't learn it ...

ACAP	TN3270	MANET/AUTOCONF	NETCONF
APEX	URI,URL,URN issues	MobileIP	POLICY
ATOM	VoIP	NEMO	SNMP
BEEP	WEBDAV	NETLMM	Traffic Engineering
CALSCH	WIDEX	OSPF	DIAMETER
CIP	FECFRAME	PPP	EAP
DKIM	iSCSI, iFCP	PTOMAINE	IDX
DNS	MIDCOM, STUN	PWE	IEPREP, ECRIT
EDIINT	ONCRPC	RIP	INCH
Email and MIME	RDDP	Router Discovery	IPSEC, IKE
ENUM	ROHC	RSVP, Integrated Services,	KERBEROS and GSS-API
FAX	RMT	NSIS	KEYPROV
FTP	RTP,RTSP,SDP	SOFTWIREs	LTANS
GEOPRIV	SCTP	UDLR	NEA
HTTP	TCP	VRRP	OPENPGP
Instant messaging	UDP	ZEROCONF	OPSEC
IPP	BEHAVE	16ng (IP over IEEE 802.16)	OTP
LDAP	BFD	6lowpan (IPv6 over 802.15.4)	PANA
Language Tags	BGP	GMPLS	PKI
Multimedia	DHCP	IP over X	RADIUS
NFS	DIFFSERV, PCN	IPoIB	RPSEC, SIDR
NNTP	FORCES	IMSS	SACRED
NTP	GROW	MPLS	SASL
OPES	HIP	TRILL	SEND
RSERPOOL	ICMP	ANCP	SOCKS
SEAMOBY	IPv4	BMWG	SSH
SIP, SIPPING, PPSIP	IPv6	CAPWAP	SSL/TLS and HTTPS
SLP	IPMTUD iscovery	COPS	SYSLOG
TELNET	IP multicast	GSMP	S/MIME
TFTP	IS-IS	IPFIX, PSAMP	XMLDSIG
TIP	L2VPN, L3VPN	IPPM	
		MIBs	

4

Protocol stack

SLP
DNS
DHCP

Routing
OSPF, BGP4,
IS-IS, RIP

SIP
VoIP

LDAP

FTP

SNMP

Email:
SMTP, POP3,
IMAP, PGP,
S/MIME

etc., etc.

NFS

SSH Login,
telnet

Web
Services
SOAP

HTTP
HTTPS

Socket interface, secure transports (SSL, TLS, SSH, SASL)

RDMA

TCP, UDP, SCTP, DCCP, RTP

QOS: Diffserv, RSVP

ROHC

Mobile IP, multicast

IPv4, IPv6, ICMP, ARP

Security: IPsec, IKE

RADIUS, DIAMETER

Layer 3 VPNs, SOCKS, NAT

EAP authentication

Link layer encapsulation, packet drivers
IP over X, PPP, MPLS, Layer 2 VPNs

Link layer (Ethernet, 802.11, ADSL, T1, SONET...)

Topology

A

B

← Autonomous Systems

Area

DNS, DHCP servers

Site OSPF

R1

ISP2

Back bone

R3

Firewall

BGP4

R21

DNS, DHCP servers

Local ISP

ISP1

R99

NAT+firewall

ADSL user

6

The end-to-end principle (1)

- Note how TCP works - it *assumes* that packets may be lost, delayed, corrupted or delivered out of order. The two ends of a TCP connection cooperate to overcome this

- Note how SSH works - it *assumes* that messages may be intercepted and that attackers may try to insert false messages. The two ends of an SSH connection cooperate to overcome this

- Note how DNS works - if a DNS (UDP) message is lost, no harm results except a delay.

- These are all examples of the end-to-end principle at work

The end-to-end principle* (2)

- Certain required end-to-end functions can only be performed correctly by the end-systems themselves

- Any network, however carefully designed, will be subject to failures of transmission at some statistically determined rate. The best way to cope with this is to give responsibility for the integrity of communication to the end systems. A similar argument applies to intrusions

- No solution buried inside the network can give the same level of assurance as the end systems

  – For example, *end-to-end* encryption is intrinsically safer than *router-to-router* encryption

see References

8

Other principles (1)

- Heterogeneity by design

- Avoid duplicate solutions

- Scaleable designs

- Performance and cost must be considered as well as functionality

- KISS (keep it simple, stupid!)

- Modularity is good

- Good enough is enough (don't seek perfection)

- Minimise use of options

- Be strict when sending and tolerant when receiving

Other principles (2)

- Be parsimonious with unsolicited packets, especially multicasts and broadcasts

- Circular dependencies must be avoided

- Objects should be self-decribing (type and size)

- Nothing gets fully standardised until there are multiple instances of running code

- Avoid design that requires hard coded addresses

- Addresses must be unambiguous (NAT breaks this!)

- Designs should be fully international

- All protocols need strong security (early ones didn't!)

References

- RFC 1958: Architectural principles of the Internet

  - End-to-end principle paraphrased from "End-To-End Arguments in System Design", J.H. Saltzer, D.P.Reed, D.D.Clark, ACM TOCS, Vol 2, Number 4, 1984

- "Why the Internet only just works" by Prof. Mark Handley, University College London

  http://www.cs.ucl.ac.uk/staff/ M.Handley/papers/only-just-works.pdf

314 s1 Exam, 2008

- Exam Date: Saturday, 28 June 2008, at Tamaki

- Time: 9:15 - 11:30 am

- 10 *short*-answer questions

- Material covered includes

  - lecture slides

  - assignments

Questions?

- What haven't you understood in this course?