

# Computer Networks

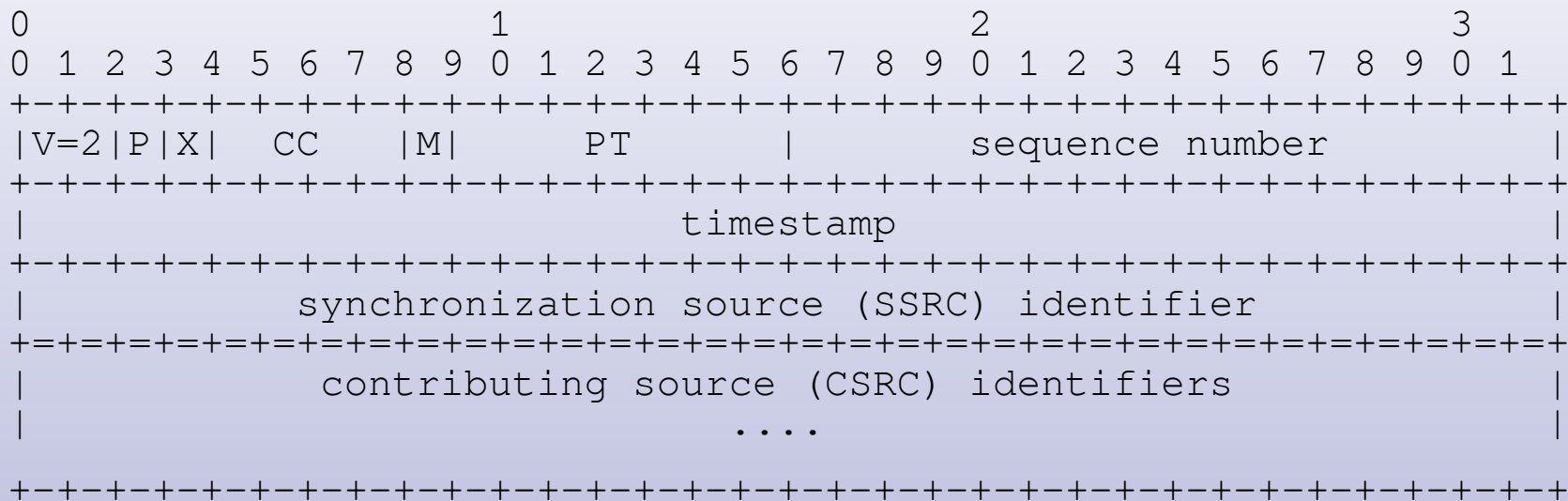
## ICS 651

- Real Time Protocol (RTP) and RTP Control Protocol (RTCP)
- peer-to-peer (P2P) networks
- bittorrent
- Distributed Hash Tables, DHTs
- bitcoin
- allnet

# Real-Time Protocol, RTP

- for time-sensitive data where loss is acceptable
- defined as two protocols, RTP and RTP Control Protocol or RTCP (RFC 3550)
- RTCP provides quality of service and congestion feedback
- an RTP profile is defined for video and audio (RFC 3551)
- typically layered on top of UDP, with an even port number for RTP and the next higher port number for RTCP

# Real-Time Protocol, RTP



- P records whether the data is padded
- X records whether there is a header extension
- CC is the number of contributing source identifiers
- M is for application purposes in marking within the stream
- PT is profile-dependent, identifies media type
- sequence number is for application purposes
  - packets may not be delivered in sequence
- timestamp identifies the time of the sample
- synchronization source is a randomly chosen ID for each source of timestamps
- contributing sources are the SSRC IDs of the sources that have contributed to this data

# Additional Real Time Protocols

- Real Time Streaming Protocol, RTSP, controls one or more sources of real time data, e.g. RTP streams (RFC 2326)
- Session Initiation Protocol provides additional session-level primitives, (RFC 3261)

# Real-Time Audio and Video transmission

- original digital telephone network designed to carry 8,000 voice samples per second per channel: synchronous network (SONET)
- compressed audio and video still need timely delivery, but do not have bandwidth guarantees
- audio and video may be multicast for conferencing
- data loss and data delay have the same effect
- for pre-recorded data, if maximum delay is known, can pre-stream and buffer a corresponding number of frames (Video On Demand, VoD)
- actual bandwidth, delay, and jitter experienced on stream define the Quality of Service, QoS
- a QoS may also be requested as part of resource reservation
- if audio stream is given higher priority,
  - the overall transmission can tolerate higher data loss, and
  - temporary loss of video stream is less disruptive than temporary loss of audio stream
- video and audio streams must be re-synchronized at receiver

# Peer-to-peer (P2P) networks

- moving away from client-server model:
  - everyone has both client and server functionality
    - if using TCP, still has accept/connect distinction
  - nobody is authoritative
  - as decentralized as possible
- logically every participant is both client and server, can both connect and be connected to
- often special-purpose, still no general-purpose P2P network
- usually overlay networks built on top of the existing Internet protocols

# BitTorrent

- P2P content delivery/distribution network for files
- files may be large: bittorrent supports requesting and delivering parts of files
- each part of a file is identified by a hash
- torrent file lists all the hashes for a file
- a peer that has downloaded a part of a file can forward it to other peers
- a **tracker** node keeps track of peers that have parts of files

# BitTorrent Reliability

- initial distribution (seeding) must succeed for content to be available
- at least one tracker must be available to peers for download to be possible
- popular files are downloaded from several peers at once, maximizing bandwidth and minimizing the load on any one peer
- two peers can exchange different parts of the same file
  - peers can preferentially send content to peers from which they receive content
- unpopular files may disappear and no longer be available



# Distributed Hash Table Goals

- DHT
- P2P self-organizing storage network – and sometimes also a communications network
- access to one node of the network should give access to the entire content of the network
- data is identified by keys which are fixed-length bitstrings

# DHT ID space and routing

- each node selects a random  $k$ -bit ID
- each node keeps a "routing table" with up to  $k$  entries
- the  $i$ -th entry in the routing table, if any, has a route to a node that has the same first  $i-1$  bits of the ID, and a different  $i$ -th bit
- in this way, the first entry in the routing table is in the other half of the ID space
- the second entry in the routing table is in the same half of the ID space, but the other quarter, and so on
- to route a packet from node  $x$  to ID  $y$ , where the first  $i$  bits of  $x$  and  $y$  are the same, the packet is sent to the node (if any) in the  $i$ th position of the routing table -- this node will be closer to the destination
- if the network has  $N$  nodes, the routing table only needs  $\log N$  entries

# DHT storage

- each node stores data for its own ID, and all IDs up to  $n-1$  nodes later in ID space (maybe  $n=4$ )
- a request is routed until there is no route to a closer node. At this point, the requested data should be on the node
- when nodes leave the network (**churn**), the data should still be available redundantly in other nodes -- a data redistribution scheme then tries to have every item in at least  $n$  nodes

# DHT performance

- at least one node must be reachable in order to connect with the network
- messages for new data must go through an average of  $\log N$  hops, where each hop may cross the entire Internet
- when there is a lot of churn, routing tables must be recomputed frequently and data stored in the network must be redistributed frequently

# Bitcoin

- distributed, reliable ("hard to falsify") time-stamping network
- each time-stamp record ("block") includes existing transactions, and a summary (hash from Merkle tree) of all previous transactions
- finding valid blocks require compute power,
  - first n bits of hash must all be zero
  - can only be found at random (we think!)
  - n evolves to keep the number of blocks generated at a near constant rate
- "block miners" get a reward for finding a block with a suitable hash
- each block keeps track of all transactions since the last block
- P2P network distributes the blocks to all peers

# Bitcoin transactions

- each transaction has a list of input transactions, adding up to an amount  $B$
- each transaction has a list of output amounts and public keys, adding up to an amount  $B' \leq B$
- if  $B' < B$ , the difference  $B - B'$  belongs to the block miner as a transaction fee
- the input transactions must provide evidence of having a private key equivalent to the public key of the corresponding output transaction

# Bitcoin block chain

- each bitcoin miner produces and hashes as many blocks as possible until it finds a hash with the first n bits zero
- the miner then broadcasts the block as fast as possible
- every other miner includes this block into a new block they try to generate
- two (or more) miners may find blocks A and B approximately simultaneously, and broadcast both throughout the network
- every miner has a choice of which block to include in the block they are working on
- the next successful block C will have picked one of the winners of the previous round, A or B
- almost all miners will now build their block on C, confirming either A or B, and one of them will produce a new block D
- eventually, only one of A or B is confirmed by the growing block chain

# Other applications of blockchains

- distributed, reliable network for time-stamping transactions
- might be useful anywhere distributed recordkeeping is needed
  - e.g. real-estate transactions
- but proof-of-work is expensive!
  - proof-of-stake may be a less energy-intensive alternative
  - bankers may like the idea of a distributed network limited to only carefully-selected participants
- in any case, there must be incentive for the miners
  - e.g. in bitcoin, miners get newly-minted coins and transaction fees



# AllNet

- See this 4-minute talk at the March 30th, 2016, Wetware Wednesday

<http://www2.hawaii.edu/~esb/2016spring.ics651/allnet-talk.pdf>

- See also

<http://alnt.org/>

# AllNet status, 2018

- ad-hoc communication between Linux systems of ad-hoc 802.11 WiFi
- DHT used for communication over the Internet
- bandwidth-limited to an average 8KB/second
  - priorities let us forward own packets even when bandwidth is limited
- trace mechanism for debugging, similar to ping+traceroute

b1.01/16	0 hop	
b3.00/16	1 hop	about 50m away!
b4.00/16	2 hop	
b5.00/16	3 hop	2.519748s rtt