Exercise
for Lecture “P2P Systems”

Prof. Dr. David Hausheer
Dipl.-Wirtsch.-Inform. Matthias Wichtlhuber, Leonhard Nobach, M. Sc., Dipl.-Ing. Fabian Kaup, Christian Koch, M. Sc., Dipl.-Wirtsch.-Inform. Jeremias Blendin

TECHNISCHE UNIVERSITÄT DARMSTADT

Summer Term 2015
Exercise No. 12
Published at: 07.07.2015, Submission date: 14.07.2015
Submission only via the Moodle platform in PDF, plain text, or JPG/PNG.
Contact: `[mwichtlh|lnobach|fkaup|ckoch|jblendin]@ps.tu-darmstadt.de`
Web: `http://www.ps.tu-darmstadt.de/teaching/p2p/`
– Example Solution –

Problem 12.1 - BitTorrent

A) In the lecture, you have learnt about the incentive applied by BitTorrent. Assume two peers having the choice to cooperate or defect. Uploading data has a cost of $C_F = 1$ and downloading has a slightly higher value $U_F = 1 + \epsilon$, as the peer receives content. Derive the payoff matrix, taking the properties of the unchoking algorithm into account. What is the dominant strategy?

Solution:
The unchoking algorithm creates a payoff of 0 for both sides, whenever one of the players defects. Thus, the dominant strategy is Cooperate, as it maximizes the payoff for both sides:

P_1; P_2	Cooperate	Defect
Cooperate	$R_1 = \epsilon$; $R_2 = \epsilon$	$S_1 = 0$; $T_2 = 0$
Defect	$T_1 = 0$; $S_2 = 0$	$P_1 = 0$; $P_2 = 0$

B) Discuss the trade-off between piece overlap and the number of distributed copies in a swarm. How does BitTorrent maintain a good balance?

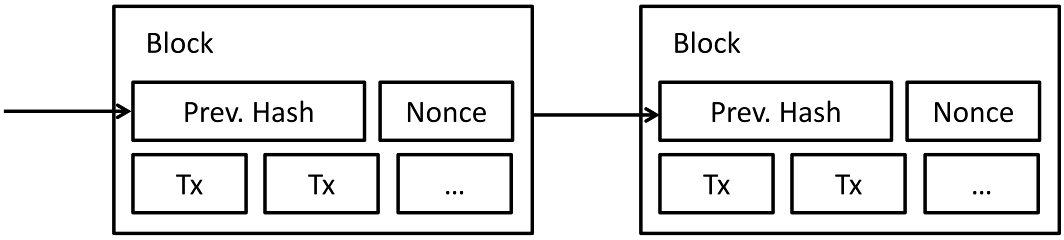
Solution:
Piece overlap: A low piece overlap maxmimizes the possibilities for trading pieces.
Distributed copies: A high number of distributed copies ensures, that chunks are not lost, when a peer goes offline.
BitTorrent maintains a good balance of piece overlap and distributed copies by applying a rarest-first piece picking strategy.

C) Derive a formula for the average number of requests needed to receive a new chunk out of n pieces, if you already possess $m \leq n$ pieces. Use this formula as a base to define the average number of requests for receiving all pieces. Calculate the average number of requests needed for $n = 7$ pieces.

Solution:
Requesting $m + 1$th piece of n pieces: $S(m) = \frac{n}{n-m}$

Requesting all n pieces: $S = S(0) + S(1) + \ldots + S(n) = \frac{n}{n} + \frac{n}{n-1} + \ldots + \frac{n}{1} = n * \sum_{k=1}^{n} \frac{1}{k}$
Average number of requests for $n = 7$ pieces: $S = \frac{363}{20} = 18.15$

Problem 12.2 - Bitcoin



A) Bitcoin applies a cryptographic puzzle as a proof of work. The peer guesses a nonce in a hash function H, until the following (slightly refined) condition holds: $H(H(\text{previous hash}), \text{transactions}, \text{nonce}) \leq 2^n$. Assume the length of the hash to be 128 bits and the difficulty $n = 120$. Calculate the average number of guesses needed to solve the block with a probability of 99%.

Solution:
Probability of finding a matching nonce: $P(\text{hit}) = \frac{2^n}{2^{128}}$
Probability of not finding a matching nonce: $P(\text{no hit}) = 1 - \frac{2^n}{2^{128}}$
Probability of finding a match after i guesses with probability of 99%: $1 - P(\text{no hit})^i = 0.99$
Case $n = 120$:
$1 - (1 - \frac{2^{120}}{2^{128}})^i = 0.99 \Leftrightarrow$
$1 - (1 - 2^{120-128})^i = 0.99 \Leftrightarrow$
$1 - (1 - 2^{-8})^i = 0.99 \Leftrightarrow$
$(1 - 2^{-8})^i = 0.01 \Leftrightarrow$
$i = \log_{1-2^{-8}} 0.01 \Leftrightarrow$
$i = 1176.62$ guesses

B) Discuss the scalability of Bitcoin with respect to the block chain concept and the way new transactions and solved blocks are spread in the network.

Solution:
Block chain: The block chain has to be downloaded and verified in advance, before a peer can join the network. It contains all transactions ever performed in the network. Thus, scalability is doubtful.
Networking: All communication is done via flooding (new transactions/solved blocks). Thus, Bitcoin will suffer from the same problems (message storms) as other flooding based protocols (e.g. Gnutella 0.4), if the number of participants increases.