Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing

Systems@ETH Zürich

Prof. T. Roscoe, Prof. R. Wattenhofer

Computer Systems

— Solution to Assignment 12 —

1 Game Theory

Quiz

1.1 Selling a Franc

We assume that there are two bidders, b_1 and b_2. If b_1 bids 5 rappen, his gain is 95 rappen. Now b_2 is inclined to bid 10 rappen and gains 90 rappen. This can continue until b_1 bids 95 rappen. Bidder b_2 now has the choice of losing 90 rappen (her last bid) or coming out even. Since she is a rational player, she will bid 1 franc. Bidder b_1 now faces a similar choice. Either he loses 95 rappen or he bids and has a chance of only losing 5 rappen. Since he is a rational player, he will bid 1.05 franc. This bidding war will continue indefinitely (or until one bidder runs out of money).

There are a few ways the bidders could have avoided this situation. Apart from the obvious, simply do not play, they could have also colluded. One bidder bids 5 rappen for the franc and the bidders will simply split the money they made. This requires that the bidders can trust each other. As you can guess, there are games that anticipate collaboration.
There exists however a strategy, which is profitable even for a non-colluding bidder. If the first bidder bids 95 rappen, she will win 5 rappen, because nobody else will also bid. Why will nobody else bid? If another bidder bids more, he will certainly not bid more than 1 franc, because this will yield a negative payoff. Instead, he could bid 1 franc. But then, the first bidder will bid 1.05 francs to minimize her loss. And then the game continues as outlined at the beginning and both bidders will incur a loss. Therefore, no rational player will bid 1 franc in this scenario.

Basic

1.2 Selfish Caching

To be sure that we find every Nash Equilibrium, we explicitly write down every best response.

i. The best response strategies are

u: cache only if nobody else does. (B1)

v: cache if neither u nor x cache. (B2)

w: cache unless u caches. (B3)

x: cache if neither u nor v cache. (B4)

Nash equilibrium. If we assume that u plays $Y_u = 1$ (u caches) the system can only be in a NE if $Y_v = Y_w = Y_x = 0$ due to (B1). Since for all v, w, and x it is the best response not to cache if u does, $x = (1000)$ is an Nash equilibrium. If $Y_u = 0$ then (B3) implies $Y_w = 1$. If furthermore, $Y_v = 1$ it must hold that $Y_x = 0$ due to (B2). This does not conflict with (B4), and (0110) constitutes another NE. Last, if $Y_v = 0$ then (B2) implies $Y_x = 1$, which is also okay with (B4). Hence (0011) is also a NE.

$$NE = \{(1000), (0110), (0011)\}$$

Price of anarchy. The social optimum is achieved in strategy profile (1000), namely $OPT = cost(1000) = 1 + \frac{1}{2} + \frac{3}{8} + \frac{3}{4} = \frac{21}{8}$. Since (1000) is also a Nash equilibrium we immediately get that $OPoA = 1$. The worst-case price of anarchy is

$$PoA = \frac{cost(0110)}{OPT} = \frac{\frac{1}{2} + 1 + 1 + \frac{7}{8}}{\frac{21}{8}} = \frac{9}{7} \approx 1.286.$$

ii. The best response strategies are

u: cache only if nobody else does. (B1)

v: cache unless u caches. (B2)

w: cache unless x caches. (B3)

x: cache if neither u nor w cache. (B4)

Nash equilibrium. If we assume that u plays $Y_u = 1$ (u caches) the system can only be in a NE if $Y_v = Y_w = Y_x = 0$ due to (B1). However, $Y_x = 0$ implies that $Y_w = 1$ due to (B3), and hence there can be no NE with $Y_u = 1$. In any NE it must hold that $Y_u = 0$. Consequently, it must hold that $Y_v = 1$ from (B2). Now if $Y_w = 1$ (B3) implies that x does not cache. This does not infringe rule (B4), and thus $x = (0110)$ is a Nash equilibrium. If $Y_w = 0$ then (B4) implies that x caches. As thus, rule (B3) is not violated $x = (0101)$ is also a Nash equilibrium.

Price of anarchy. The social optimum is achieved in strategy profile (0110), namely $OPT = cost(0110) = \frac{1}{3} \cdot 0.2 + 1 + 1 + \frac{1}{2} \cdot 0.2 = 2.1\overline{6}$. Since (0110) is also a Nash equilibrium we get that the optimistic price of anarchy is 1. The worst-case price of anarchy is

$$PoA = \frac{cost(0101)}{OPT} = \frac{1/3 \cdot 0.2 + 1 + 0.2 + 1}{2.1\overline{6}} = \frac{68}{65} \approx 1.046$$

1.3 Selfish Caching with variable caching cost

We define D_i to be the set of nodes that cover node i. A node j *covers* node i if and only if $c_{i \leftarrow j} < \alpha_i$, i.e., node i prefers accessing the object at node j than caching it. Convince yourself that a strategy profile is a Nash Equilibrium if and only if for each node i it holds that

- if $Y_i = 1$ then $Y_j = 0$ for all $j \in D_i$, and
- if $Y_i = 0$ then $\exists j \in D_i$ with $Y_j = 1$.

i. $D_u = \emptyset$, $D_v = \{u, w\}$, $D_w = \{u\}$. D_u being empty implies $Y_u = 1$ (i.e. caches the file). Hence $Y_v = 0$, and $Y_w = 1$. $NE = \{(101)\}$. $PoA = 1$ since (101) is also the social optimum strategy.

ii. $D_u = \{v\}$, $D_v = \{u\}$, $D_w = \{u, v\}$. If $Y_u = 1$, then $Y_v = 0$ and $Y_w = 0$. If $Y_u = 0$, then $Y_v = 1$. Hence $Y_w = 0$. The equilibria are $NE = \{(100), (010)\}$.

$$PoA = \frac{cost(100)}{cost(010)} = \frac{3 + 1 + 8/3}{3/2 + 3/2 + 5/3} = \frac{40}{28} \approx 1.43$$

Dominant strategies. Every dominant strategy profile is also a Nash equilibrium. Hence we only have to check the computed NEs whether they consist of dominant strategies only.

Let us consider game i. Since every dominant strategy profile is also a Nash Equilibrium, it suffices to consider the NE. The game has no dominant strategy profile. Profile (101) is no dominant strategy profile in game i. since, although $Y_u = 1$ is the dominant strategy for u, $Y_v = 0$, and $Y_w = 1$ are not dominant strategies for v and w. If $Y_v = 1$, then it would be the best response of w to set $Y_w = 0$. Game ii: Since the decision of node u whether to cache depends on the decision of node v, this is not a dominant strategy. Therefore, this game has no dominant strategy profile.

Advanced

1.4 Matching Pennies

The bi-matrix of the game with Tobias as row player, and Stephan as column player looks as follows:

	H	T
H	1 , -1	-1 , 1
T	-1 , 1	1 , -1

This zero-sum game has no pure Nash equilibrium. For the mixed NEs, Tobias plays heads (H) with probability p, tails (T) with probability $1-p$. Stephan plays H with probability q, and T with probability $1-q$. We get the expected utility functions Γ:

$$\begin{aligned}\Gamma_T(p,q) &= p(q-(1-q)) + (1-p)(-q+(1-q)) = (4q-2)\cdot p + 1 - 2q \\ \Gamma_S(p,q) &= q(-p+(1-p)) + (1-q)(p-(1-p)) = (2-4p)\cdot q + 2p - 1\end{aligned}$$

If Stephan plays $q = 1/2$ the term $4q-2$ equals 0, and any choice of p will yield the same payoff for Tobias. If Tobias plays $p = 1/2$ then any choice of q is a best response for Stephan. Thus $(p,q) = (1/2, 1/2)$ is a mixed NE. Note that for any choice of $p > 1/2$, Stephan's best response is to choose $q = 0$. For a $p < 1/2$ Stephan would choose $q = 1$. However, Tobias' best response to $q > 1/2$ is $p = 1$, and $p = 0$ if $q < 1/2$. Hence $(p,q) = (1/2, 1/2)$ is the only pair of mutual best responses.

1.5 PoA Classes

Let I^n be an instance of $\mathcal{A}^n_{[a,b]}$ that maximizes the price of anarchy, i.e. $PoA(\mathcal{A}^n_{[a,b]}) = PoA(I^n)$. Let $x, y \in X$ be two strategy profiles in I^n such that $PoA(I^n) = cost(y)/cost(x)$. We show the claim by constructing an instance $\hat{I}^n \in \mathcal{W}^n_{\left[\frac{1}{b},\frac{1}{a}\right]}$ out of I^n for which it holds that $PoA(\hat{I}^n) \geq \frac{a}{b} PoA(I^n) = \frac{a}{b} PoA(\mathcal{A}^n_{[a,b]})$. We construct $\hat{I}^n$ by setting $d_i = 1/\alpha_i$, $\hat{\alpha}_i = 1$ where α_i are the placement costs (for local caching) of player i in I^n. All edges remain as in I^n. This game has the same Nash equilibria as I^n since the cover sets D_i (nodes for which we do not cache if these cache already) for each peer stay the same. A peer j is in D_i iff $c_{i \leftarrow j} < \alpha_i$, or $c_{i \leftarrow j}/\alpha_i < 1$ respectively. We get the bound by comparing the performance of the two strategies x, y that produce the PoA in I^n in $\hat{I}^n$. Note that x is not necessarily a social optimum in $\hat{I}^n$, but y is a Nash equilibrium

also in $\hat{I}^n$, because the cover sets are the same.

$$
\begin{aligned}
PoA(\hat{I}^n) &\geq \frac{\hat{cost}(y)}{\hat{cost}(x)} = \frac{\sum_{i=1}^{n}\left(y_i + (1-y_i)\frac{c_i(y)}{\alpha_i}\right)}{\sum_{i=1}^{n}\left(x_i + (1-x_i)\frac{c_i(x)}{\alpha_i}\right)} && (1)\\
&= \frac{b \cdot a\sum_{i=1}^{n}\left(y_i + (1-y_i)\frac{c_i(y)}{\alpha_i}\right)}{b \cdot a\sum_{i=1}^{n}\left(x_i + (1-x_i)\frac{c_i(x)}{\alpha_i}\right)} && (2)\\
&\geq \frac{a\sum_{i=1}^{n}\left(y_i\alpha_i + (1-y_i)c_i(y)\right)}{b\sum_{i=1}^{n}\left(x_i\alpha_i + (1-x_i)c_i(x)\right)} && (3)\\
&= \frac{a \cdot cost(y)}{b \cdot cost(x)} = \frac{a}{b}PoA(I^n) && (4)
\end{aligned}
$$

$\hat{cost}(x)$ denotes the cost function in $\hat{I}^n$. x_i, and y_i are either 1 or 0. x_i equals 1 if player i caches in strategy profile x, and 0 if she does not. With $c_i(y)$ we denote the cost of node i if it access the file remotely in strategy y. For step (3) we exploit the fact that $b \geq \alpha_i$ and $a \leq \alpha_i$ for all i.

2 Authenticated Agreement

Quiz

2.1 PBFT: Basics

a) According to Lemma 25.18, it is impossible that two prepared-certificates for the same sequence number are gathered within the same view (not even at different nodes). Therefore, once a node has obtained a prepared-certificate for a request r in view v, it can be sure that no correct node will execute a different request $r' \neq r$ with the same sequence number in view v, as this would also require a prepared-certificate for the other request within the same view.

b) The new primary has to send around the new-view-certificate $\mathcal{V}$; that certificate has to be valid and the set of `pre-prepared`-messages $\mathcal{O}$ has to be constructed validly from $\mathcal{V}$ in the way specified by the protocol. Since $\mathcal{V}$ already determines the content of $\mathcal{O}$ and the `view-change`-messages in $\mathcal{V}$ are signed, correct replicas can rely on $\mathcal{O}$ if the above conditions hold.

c) Not necessarily. It is possible that some node u collected a prepared-certificate for a triple (v, s, r), but as soon as u collected the prepared-certificate, a view change happened. In that case, no correct node can have executed that request yet, but u's `view-change`-message could still end up in the set $\mathcal{V}$ of the `new-view`-message for the next view.

d) The proof of Theorem 25.26 shows that if a request was executed by a correct node, then a prepared-certificate will end up in $\mathcal{V}$. If we take the contrapositive of that statement, we find that if there is no prepared-certificate for a request in $\mathcal{V}$, then no correct node has executed that request yet. Omitting prepared-certificates for requests that no correct node executed cannot harm correctness of the system.

Basic

2.2 PBFT: Utility of the Phases of the Agreement Protocol

a) Backups start their faulty-timer after they receive a request. If backups do not forward requests to the primary, then a faulty client could just send requests to the backups, and the backups' faulty timers would permanently keep expiring, inducing view change after view change.

A byzantine client could make sure to send a request to a backup even without knowing which node is the primary by simply sending distinct requests to all nodes; all but one node will be backups, and all of their faulty-timers would start running for requests that the primary has never seen and for which the primary can therefore not start the agreement protocol.

b) Lemma 25.18 implies that two correct nodes cannot agree to execute different requests within a single view, and the proof does not rely on nodes waiting for `commit`-messages, so this Lemma remains intact even with the alteration made in this exercise.

However, the `commit`-messages are important for the view change protocol to maintain safety across views, which we can see in the proof of Theorem 25.26. Consider the following sequence of events:

1. Node u collects a prepared-certificate matching (v, s, r), and directly executes r. No other node has seen a prepared-certificate yet, and a view change occurs at this moment.
2. The new primary p' of view $v' > v$ collects $2f + 1$ `view-change`-messages, and u's message is too slow to be included. p' thus does not add a $\texttt{pre-prepared}(v', s, r, p')_{p'}$-message to $\mathcal{O}$.
3. In the new view v', correct nodes (with the "help" of byzantine nodes) run the agreement protocol for (v', s, r') for some $r' \neq r$. As soon as correct node $w \neq u$ collects a prepared-certificate matching (v', s, r'), node w will execute r' with sequence number s.

With this, u will execute r with sequence number s, and w will execute $r' \neq r$ with sequence number s.

If $s < s^{\mathcal{V}}_{max}$ (cf. Algorithm 25.24), then r' will be `null`. However, if $s > s^{\mathcal{V}}_{max}$, then r' can be a distinct non-`null` request.

Advanced

2.3 Authenticated Agreement

a) We can do roughly the same as we did in Algorithm 25.2, but for multiple values in parallel. Every backup will be collecting messages for every value they hear about. If a correct node gathered agreement for multiple values (or for no values) after $f + 1$ rounds, then it knows that the primary must be faulty. The new algorithm can be seen in Algorithm **??**.

b) The proof is very similar to the one in the script, so we will only give a rough sketch of how to adapt it here:

- If the primary is correct, then he only sends one message $\texttt{value}(x)_p$ in the first round, and all correct backups decide on x after round $f + 1$.
- If the primary is byzantine, then there are these cases:
  1. No correct node ever adds a value to A, then all correct nodes output "sender faulty".

Algorithm 1 Byzantine Agreement with Authentication

Code for primary p:

1: $x \leftarrow$ input value of p
2: broadcast $\texttt{value}(x)_p$
3: decide x and terminate

Code for backup b:

4: $A \leftarrow \emptyset$
5: **for all** rounds $i \in \{1, \ldots, f+1\}$ **do**
6: **for all** messages $\texttt{value}(x)_u$ that b received this round **do**
7: $V_x \leftarrow \{$all messages $\texttt{value}(x)_v$ that b received since round $1\}$
8: **if** $|V_x| \geq i$ and $\texttt{value}(x)_p \in V_x$ **then**
9: $A \leftarrow A \cup \{x\}$
10: broadcast $V_x \cup \texttt{value}(x)_b$
11: **end if**
12: **end for**
13: **end for**
14: **if** $|A| = 1$ **then**
15: decide on the single element in A and terminate
16: **else**
17: decide "sender faulty" and terminate
18: **end if**

2. (The proof of this case is analogous to correct nodes deciding on 1 in the proof in the script. Check the proof in the script if some detail here is unclear.)
   At least one correct node adds at least one value x to A. For any value x that gets added to A by some correct node, the first time a correct node adds x to A necessarily happens in a round $i < f + 1$, and all correct nodes will have $x \in A$ in round $i + 1 \leq f + 1$. Since this holds for all x, all correct nodes have the same A after round $f + 1$.
   If A contains exactly one value after round $f + 1$, then all correct nodes decide on that value, otherwise all of them decide on "sender faulty".

2.4 Multiple Prepared-Certificates for the Same Sequence Number!? — How can this happen?

In Quiz question c), we saw that a correct node can collect a prepared-certificate that will not be included in a `new-view`-message. Working from this insight, we can imagine the following sequence of events:

1. Only correct node u collects a prepared-certificate for (v, s, r).
2. A view change to view $v+1$ happens, and the prepared-certificate that u collected for (v, s, r) is not included in the view change.
3. Correct node w collects a prepared-certificate for $(v + 1, s, r')$ for some $r' \neq r$.
4. A view change to view $v + 2$ happens. Both prepared-certificates are included in the `new-view`-message.

2.5 Multiple Prepared-Certificates for the Same Sequence Number!? — How can we fix it?

Notice that step 2. of the solution to Exercise **??** can only occur if the prepared-certificate that u collected was for a request that no correct node executed, see the solution to Quiz question d). This means that we can ignore that prepared-certificate without worrying about it.

On the other hand, for every request that was executed by some correct node, a prepared-certificate for it will end up in every subsequent `new-view`-message, see the proof of Theorem 25.26.

From these considerations, we define the protocol to do the following in Algorithm 25.24: if multiple prepared-certificates for sequence number s end up in $\mathcal{V}$, then let v^* be the highest view number for which a prepared-certificate (v^*, s, r) exists in $\mathcal{V}$. The primary p of new view v will include a $\texttt{pre-prepare}(v, s, r, p)_p$-message in $\mathcal{O}$ and ignore all other prepared-certificates for s. The claim we now make is this:

Theorem 1. *Let v^* be the maximum view number of any prepared-certificate for s in $\mathcal{V}$. If the $\mathcal{V}$-component of a* `new-view`*-message contains multiple prepared-certificates for the same sequence number s, then due to Lemma 25.18, there is at most one such prepared-certificate per view v^*. While constructing $\mathcal{O}$ in Algorithm 25.24 during a view change, let the primary of the new view only include a* `pre-prepare`*-message for the prepared-certificate matching s with the highest view number among the prepared-certificates for s in $\mathcal{V}$ (this is the* latest *prepared-certificate for s). Then, if some correct node executes a request r with sequence number s in view v, then the latest prepared-certificate for s in $\mathcal{V}$ during every view change after v will match (s, r).*

Proof. If no `new-view`-message ever contains two prepared-certificates for the same sequence number s but different requests $r' \neq r$, then the Theorem is already proved in the proof of Theorem 25.26 in the script. If no correct node ever executed a request at some sequence number s, then there cannot be a problem with correctness at s either. Thus, assume that a correct node executed request r with sequence number s in view v, and some subsequent `new-view`-message contains multiple prepared-certificates for s. We prove the theorem by induction, showing that once a correct node executed r with sequence number s in view v, then all prepared-certificates for sequence number s that nodes collect in later views will match (s, r).

Base case: Consider the first view $v' > v$ in which the new primary p' sends a valid `new-view`-message. Since correct backups reject invalid messages, no correct node entered any view $v^\dagger$ with $v < v^\dagger < v'$, so no prepared-certificates were collected in any such view $v^\dagger$. Thus, v is the highest view number for which any node collected a prepared-certificate for s. As shown in the proof of Theorem 25.26, a prepared-certificate matching (v, s, r) will be in $\mathcal{V}$ in the `new-view`-message for v'.

Induction step: Consider a view change from v' to v'' with $v < v' < v''$ and assume that up to v', the latest prepared-certificate in the $\mathcal{V}$-component of `new-view`-messages for s has matched (s, r). Because backups respond to $\mathcal{O}$ in Algorithm 25.25 before responding to any other `pre-prepare`-messages, nodes can only have collected a prepared-certificate for s with the same r during view v'. Thus, during the view change from v' to v'', the latest prepared-certificate for s that is in $\mathcal{V}$ will match (s, r) as well. □

Notice how this clarifies the answer to Exercise **??**: it can happen that multiple prepared-certificates for the same sequence number s exist in a `new-view`-message. However, if the new primary always picks the latest such prepared-certificate to react to when constructing $\mathcal{O}$, then that guarantees that once a request r was executed at s by any correct node, then no node will ever be able to gather a prepared-certificate for (s, r') with $r' \neq r$. Thus no correct node will ever execute anything but r at sequence number s.