Principles of Distributed Computing
Exercise 8: Sample Solution

1 Multi-Valued Agreement

The following protocol implements asynchronous multivalued Byzantine agreement according to
Definition 9.8 (relaxed for randomized protocols). It uses digital signatures and calls a (random-
ized) protocol for binary Byzantine agreement.

Algorithm 1 Multi-Valued Agreement
1: upon propose(v):
2: send the signed message (value, v) to all servers
3:
4: upon receiving 2t + 1 messages (value, *) with proper signatures:
5: let m be the value v that occurs most often in the received value messages
6: let M be the set of received value messages
7: send the message (majority, m, M) to all servers
8:
9: upon receiving n - t messages (majority, *) with valid proofs:
10: if all values m in the received majority messages are the same then
11: let M be the majority value
12: propose 1 for binary agreement
13: else
14: propose 0 for binary agreement
15: end if
16:
17: upon deciding for b in binary agreement:
18: if b = 1 then
19: decide for M
20: else
21: decide for a default value
22: end if

The protocol satisfies the standard validity condition because if all honest servers propose the
same value, all honest servers obtain a unique m, all valid majority messages contain m, and all
honest servers propose 1 for binary agreement.

Agreement and termination follow from a standard argument and from the properties of the
binary agreement protocol.
2 Strong Agreement

The standard validity condition of Binary Byzantine agreement requires a particular outcome only if all honest servers propose the same value; but the complement is that some honest server proposed the opposite value, hence any decision “makes sense” because some honest server proposed it.

Let $D$ denote the agreement domain with $m$ values and $H \subseteq D$ the set of values proposed by the honest servers. The values in $H$ are called valid. Towards a contradiction, suppose that $n \leq (m + 1)t$ and $|H| = m - 1$. Let the set of all honest servers be partitioned into $A$ and $B$ such that $|A| \leq (m - 1)t$ and $|B| = t$, such that for every $v \in H$ there are at most $t$ servers who propose $v$.

The adversary now causes all corrupted servers to follow the protocol with the invalid input $u \in D \setminus H$. The adversary isolates the servers in $B$ by delaying all messages from servers in $B$. Then the servers in $A$ must reach agreement together with the corrupted servers. But since the corrupted servers follow the protocol, they cannot be distinguished from honest servers and the protocol will decide on $u$ with some non-negligible probability. Since $u$ is not valid, this contradicts strong validity.